Physically Interpretable Neural Networks for the Geosciences: Applications to Earth System Variability

Benjamin A. Toms$^{1,*}$, Elizabeth A. Barnes$^1$, Imme Ebert-Uphoff$^{2,3}$

$^1$Department of Atmospheric Science, Colorado State University, Fort Collins, CO
$^2$Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO
$^3$Cooperative Institute for Research in the Atmosphere, Fort Collins, CO

Key Points:

- Interpretable neural networks can identify the coherent spatial patterns of known modes of earth system variability
- The layerwise relevance propagation and optimal input methods enable new ways to use neural networks for geoscientific research
- We propose that the interpretation of what a neural network has learned can be used as the ultimate scientific outcome of a trained network

*Department of Atmospheric Science, Colorado State University, Fort Collins, CO

Corresponding author: Benjamin A. Toms, ben.toms@colostate.edu
Abstract
Neural networks have become increasingly prevalent within the geosciences for applications ranging from numerical model parameterizations to the prediction of extreme weather. A common limitation of neural networks has been the lack of methods to interpret what the networks learn and how they make decisions. As such, neural networks have typically been used within the geosciences to accurately identify a desired output given a set of inputs, with the interpretation of what the network learns being used - if used at all - as a secondary metric to ensure the network is making the right decision for the right reason. Network interpretation techniques have become more advanced in recent years, however, and we therefore propose that the ultimate objective of using a neural network can also be the interpretation of what the network has learned rather than the output itself.

We show that the interpretation of a neural network can enable the discovery of scientifically meaningful connections within geoscientific data. By training neural networks to use one or more components of the earth system to identify another, interpretation methods can be used to gain scientific insights into how and why the two components are related. In particular, we use two methods for neural network interpretation. These methods project the decision pathways of a network back onto the original input dimensions, and are called “optimal input” and layerwise relevance propagation (LRP). We then show how these interpretation techniques can be used to reliably infer scientifically meaningful information from neural networks by applying them to common climate patterns. These results suggest that combining interpretable neural networks with novel scientific hypotheses will open the door to many new avenues in neural network-related geoscience research.

1 Introduction
Machine learning methods are emerging as a powerful tool in scientific applications across all areas of geoscience (e.g. Gil et al., 2018; Kapartne et al., 2018; Rolnick et al., 2019), including marine science (e.g. Malde et al., 2019), solid earth science (e.g. Bergen et al., 2019), and atmospheric science (e.g. Barnes et al., 2019; Boukabara et al., 2019; Lopatka, 2019; McGovern et al., 2017, Reichstein et al., 2019). This revolution in machine learning within the geosciences has been spurred by the coincident introduction of novel algorithms, an influx of large quantities of high-quality data, and an increase in computational power for processing immense quantities of data simultaneously. There have been limitations to the application of machine learning methods within geoscience, however, as their interpretation is commonly deemed difficult, if not impossible. Here, we show that two recent techniques from computer science for interpreting one of the most common forms of machine learning methods, Artificial Neural Networks (ANNs; hereafter just neural networks), have the potential to transform how geoscientists use neural networks within their research. More specifically, these methods enable the usage of neural networks for the discovery of physically meaningful relationships within geoscientific data.

Neural networks, also occasionally dubbed “deep learning” (LeCun et al., 2015), are one of the most versatile types of machine learning methods and can be used for a broad range of applications within the geosciences. Such models have been used for time-series prediction (e.g. Feng et al., 2015; Gardner & Dorling, 1999), identifying patterns of weather and climate phenomena within observations and simulations (e.g. Barnes et al., 2019; Gagne II et al., 2019; Lagerquist et al., 2019, Toms et al., 2019), and parameterizing sub-grid scale physics within numerical models (e.g. Bolton & Zanna, 2019; Brenowitz & Bretherton, 2018; O’Gorman & Dwyer, 2018; Rasp et al., 2018). The structure of the neural networks employed within these applications can vary substantially, although the
The general concept is the same: given a set of input variables, the neural network is tasked with identifying the desired output as accurately as possible.

Neural networks consist of consecutive layers of nonlinear transformations and adjustable weights and biases. The mathematics of how these layer-to-layer transformations are applied to the data are well understood since the individual transformations themselves are mathematically simple. However, once a neural network has been trained, the reasoning of how and why it combines information across its weights and biases and from each transformation to the next to arrive at its ultimate output is not easily deduced, due to the potentially high complexity of the network architecture and the increasing level of abstraction in later layers of the network. Thus, in practice, neural networks are overwhelmingly used - including in geoscience - without a detailed understanding of the reasoning they employ to arrive at their output.

Even for applications where the network’s output is all that is desired, a lack of understanding of a network’s reasoning can lead to many problems. For example, the neural network can overfit to the data and attempt to explain noise rather than capturing the meaningful connections between the input and output. Additionally, within the geosciences sample sizes are typically limited, which means that the available samples might not capture the full range of possible outcomes and thereby might also not be representative of the true underlying physics driving the relationship between the inputs and outputs. In this scenario, the network may therefore fail to model the relationship correctly from a physical perspective, even if it accurately captures a relationship between the inputs and outputs given the provided training data. Thus, the ability to interpret neural networks is important for ensuring that the reasoning for a network’s outputs are consistent with our physical understanding of the earth system.

The various applications of neural networks within the geosciences commonly rely on indirect scientific inference. In many cases, the primary objective of the neural networks has been to maximize the accuracy of the networks’ outputs, from which indirect inferences have been made about the earth system. For example, by using neural networks to predict the likelihood that a convective storm would produce hail, Gagne et al. (2019) showed that the neural networks made accurate predictions by identifying known types of storm structures. In another case, Ham et al. (2019) used a neural network to predict the evolution of the El Niño Southern Oscillation (ENSO), and then used interpretation techniques to show that ENSO precursors exist within the South Pacific and Indian Oceans. However, even in these cases, the primary objective was to construct a neural network that most accurately predicted its output, with the interpretation being used to ensure the network attained high accuracy using reasoning consistent with physical theory. This theme is common throughout geoscientific applications of neural networks: the network’s output is the ultimate objective, and interpretation techniques are used to ensure the network is making decisions according to our current understanding of how the earth system evolves.

We propose an additional use for neural networks, whereby the ultimate scientific objective of using a neural network is its interpretation rather than its output. From this perspective, we show how neural networks can be used to directly advance our understanding of the earth system. To do so, we focus on two methods which trace the decision of a neural network back onto the original dimensions of the input image, and thereby permit the understanding of which input variables are most important for the neural network’s decisions. These methods are particularly well-suited for scientific inference when a physical understanding of relationships is important – such as within geoscience – and one of the methods has yet to be introduced to the geoscientific community to the best of our knowledge.

We first discuss the theory and logic behind the two interpretation methods, then provide two examples of how these methods can be used to explore physically meaning-
ful patterns of earth system variability. The objective of this paper is to showcase the utility of using neural network interpretations for scientific inference. We use commonly studied climate phenomena to do so, and are optimistic that these concepts can be extended to other geoscientific applications, as well.

2 Neural Network Architecture

In this work, we use separately trained fully-connected neural networks of identical design (detailed in Figure 1). A fully-connected neural network is the most basic form of neural network. Each neural network that we use has an input layer which receives the input sample, two intermediate “hidden” layers of nodes with eight nodes each, and an output layer with two nodes. The inputs for our examples are vectorized maps (i.e. images) of geospatial phenomena and are labeled with a two-unit vector that describes which of two categories, or classes, the image is associated with. Within the two-unit labeling vector, a 1 is placed in the index that the sample is associated with and a 0 is placed in the other. The output of the neural network is therefore also a two-unit vector which represents the neural network’s estimation of the likelihood that the input sample belongs in each class such that the output vector always sums to 1. If the neural network is more confident that a sample belongs in a particular class, then the output for the corresponding unit of the output vector will be closer to 1. The objective of the neural network is to output a two-unit vector that is as similar to the label vector as possible, which means it is tasked with maximizing its confidence that each input sample belongs in its labeled category. More extensive details of the neural network architecture and training procedure are provided in the appendix.

It is worth noting that we use a basic form of a neural network for our examples, but could have chosen more advanced architectures such as convolutional neural networks (CNNs, e.g. Kriszhevsky et al., 2012). However, the intent of this paper is to present the usage of the interpretation of neural networks as a tool for scientific inference and not to showcase the utility of various neural network architectures. We therefore opt to keep the networks as simple as possible. In addition, we will show that this basic network architecture is sufficient to capture the known relationships between the inputs and outputs of our examples. The interpretation methods we use also place some restrictions on the structures of the neural networks, the details of which are discussed in the subsequent sections, and so our neural networks abide by these requirements. With that said, the interpretation methods we discuss here are also applicable to a variety of other neural network architectures.

3 Neural Network Interpretation Methods

3.1 Optimal Input

The technique we call “optimal input”, but which is also referred to as backwards optimization or feature visualization, calculates the input that maximizes a neural network’s confidence in its output (Olah et al., 2017; Simonyan et al., 2013; Yosinski et al., 2015). This method offers insights into which patterns the neural network thinks are most associated with a particular output by using the weights and biases of a trained neural network to iteratively update an input sample until it is most closely associated with a user-specified output of the network.

Once a neural network is trained, the weights and biases can be frozen, which means that they are no longer updated as the neural network sees new samples. So, in turn, the optimal input method takes the reverse approach to how a neural network is trained, and rather than updating the weights and biases of the network itself, an input sample is iteratively updated given a trained neural network with frozen weights and biases. The fact that the optimized input has the same dimensions as the samples used to train the
network is particularly useful and is helpful for determining which patterns within the input vector are most important for describing any relationships between the input and output variables. The optimized input can also be interpreted in the same units as the input samples used to train the network.

The optimal input method is illustrated in Figure 2, detailed in code in the supporting information, and proceeds as follows:

**Method Input**: User-defined output of a trained neural network  
**Method Output**: An optimized input that shows the input pattern most closely associated with the user-defined output according to the trained neural network  
**Procedure:**

1. A neural network is trained, and the weights and biases are frozen, which means that they are not updated when a sample is input into the neural network.  
2. A desired output from the neural network is defined. For example, if the network is trained to identify whether a sample belongs in one of two categories, the desired output could be when the neural network is 100% confident that the input belongs in one of the two categories.  
3. A sample is generated of the same shape as the samples used to train the neural network, but the sample is initialized as all zeros.  
4. This all-zero sample is passed through the network, and the output is gathered. The output is then compared to the desired output, and the loss (i.e. error) of the all-zero sample is calculated with respect to the desired output. The loss function is the same function used to train the network.  
5. The loss is translated backwards through the neural network to the input layer using backpropagation. But, rather than updating the weights and biases of the network along the way, the input sample itself is updated in a manner which reduces the loss using an increment of the information, or gradient, that was translated back to the input layer.  
6. Iterate over steps 4 and 5 until the input is optimized such that iterations no longer reduce the error of the neural network’s output.

Gagne et al. (2019) and McGovern et al. (2019) provide other examples of how the optimal input technique has been used in geoscience, and more specifically meteorology. We note that other techniques for the initialization of the unoptimized input sample have been suggested, such as using Gaussian noise rather than all zeros, but we have found that the optimized patterns are not sensitive to these initialization techniques for our examples.

As will be discussed throughout the remainder of this paper, the optimal input technique offers valuable insights into a neural network’s decision-making process, but it is not without its limitations. Briefly, the optimized input offers one composite perspective of the patterns the network looks for within the input data. This composite perspective introduces problems when applied to domains where, for example, multiple modes of variability may lead to the same outcome. In these cases, the optimal input may contain a combination of each mode, but will not elucidate how these modes may evolve either independently or in tandem with each other. For this reason, it is useful to also understand what information within each input sample is important for the neural network’s associated output. Fortunately, there are methods for interpreting a neural network in this manner, one of which is called layerwise relevance propagation, which we discuss next.

### 3.2 Layer-Wise Relevance Propagation (LRP)

In contrast to the optimal input technique which generates a single optimized input given a desired output, layerwise relevance propagation (LRP) considers one input
sample at a time. The form of LRP that we use was introduced to the computer science community by Bach et al. (2015). This form of LRP is also referred to as a “deep Taylor decomposition” of the neural network because of its relationship to Taylor series expansion (Montavon et al., 2017), although the more general class of methods is referred to as LRP and we will therefore refer to the method as such.

For each input sample, LRP identifies the relevance of each input feature for the network’s output, and therefore helps isolate which input features are important for a network’s output on a sample-by-sample basis. For example, if the input is an image, the resulting output from LRP is a heatmap in the dimensions of the original image that shows the regions of the image which are most important for generating the network’s output for that particular sample. It bears repeating that the heatmap is specific to the input sample and so different inputs yield different heatmaps, the patterns of which depend on how the information from that input is transferred through the network as it makes its decision. LRP can be applied to any sample that is of the same dimensions as those used to train the network, even if the neural network did not see the sample during training.

Next, we generally describe how the method traces the reasoning of a neural network’s decision process, although we refer the reader to the manuscripts of Bach et al. (2015) and Montavon et al. (2017) for more details. We note that while the LRP methods presented by Bach et al. (2015) and Montavon et al. (2017) are one formulation of LRP, new formulations can be developed according to the more general guidelines posed within Bach et al. (2015).

The algorithm of LRP is illustrated in Figure 3 and proceeds as follows:

**Method Input:** An input sample

**Method Output:** The relevance of each feature within the input sample for the associated output of the neural network

**Procedure:**

1. A neural network is trained, and the weights and biases are frozen, which means that they are not updated when a sample is input into the neural network.
2. A sample is then input into the frozen neural network, and the output values are retained. If the neural network has categorical output and uses a softmax operator following the output nodes, then the output values prior to the softmax operator are retained. A single node of the output layer is identified as the node for which the relevance should be calculated. For cases of categorical output, this node is typically the one with the highest output likelihood for the given sample.
3. The output value of the single node is then propagated backwards through the network using information about the weights and biases of each node of the neural network. The propagation is done according to a particular set of propagation rules. These rules depend on the types of the neural network and input data, and what type of information is to be inferred from the network.
4. This backwards propagation through the network is done until reaching the input layer. The resulting values have the same dimensions as the input and correspond to the relevance of each input feature for the neural network’s decision of its output.
5. This process is completed for each sample of interest, from which the relevances for each sample can be studied independently or through composites or clusters of similar patterns of relevance.

An important aspect of LRP is the rules by which the relevance is translated backwards from the output layer toward the input layer. For our purposes, we only show the relevance propagation rules that are most fundamental to the theory of LRP. The rules that we use here, and which were introduced by Bach et al. (2015), have been constructed...
such that the total summed relevance after propagation back to the input layer is equal to the value of the output. For these rules, only information that positively contributes to the output is propagated backwards, and negative weights and biases are therefore ignored. That is, only information that makes the network more confident in its categorical output is propagated backwards, and information that makes the network less confident is ignored. However, there are variants of LRP that permit the inclusion of information that reduces the network’s confidence which may also be useful for network interpretability (Montavon et al., 2017).

We note again that LRP traces information for a single output node (Bach et al., 2015). So, in the case of categorical output as we present within this paper, the relevance is propagated backwards for one of the categorical output nodes – typically the node with the maximum output likelihood for the sample of interest. If the neural network uses a softmax operator in its output layer, then during the relevance calculations the softmax operator is ignored and the relevance is calculated for the network’s output prior to the softmax. The softmax operator is helpful to ensure the network converges on a solution during training, but the pre-softmax output is more useful for interpretability purposes since it is an unscaled representation of the network’s confidence in its output.

Once a sample has been input, passed forward through the network, and the output has been collected, the first step in LRP is to use the following propagation rule to pass the information backwards from the output layer to the previous layer of nodes:

\[
R_i = \sum_j a_i w_{ij}^+ + \max(0, b_j) \frac{R_j}{\sum_i a_i w_{ij}^+ + \max(0, b_j)}
\]

Within Equation 1, the \(i\) subscript represents the \(i\)-th node in the layer of the network to which the relevance is being translated backwards, the \(j\) subscript represents the \(j\)-th node in the layer of the network from which the relevance is being translated, \(R_i\) is the relevance translated backwards to the \(i\)-th node, \(R_j\) is the relevance of the \(j\)-th node, \(a_i\) is the output from the \(i\)-th node after the non-linearity has been applied when the sample is passed forward through the network, \(w_{ij}^+\) is the weight of the connection between the \(i\)-th and \(j\)-th nodes where the \(+\) signifies that only positive weights are considered, and \(b_j\) is the bias of the \(j\)-th node. The terms within this equation are illustrated schematically within Figure 3. As previously mentioned, the form of LRP that we use neglects all negative weights and biases and only traces information backwards through positive weights and biases. This rule in Equation 1 is used to propagate the relevance backwards through the network from one layer to the next, starting with the output layer and extending backwards to the first hidden layer.

There are separate rules for translating information to the input layer from the first layer of hidden nodes, the rules of which depend on whether the values of the input features are bounded or unbounded. A case where the values are unbounded is when the data is standardized and so has zero mean and unit variance, but is not necessarily restricted from varying across all real numbers. A case where the values are bounded, on the other hand, is when all the input values are normalized between 0 and 1. For the case where the input values are unbounded, the rule for translating the relevance from the first hidden layer to the input layer is:

\[
R_i = \sum_j \frac{w_{ij}^2}{\sum_i w_{ij}^2} R_j
\]

where all terms are as previously discussed for Equation 1. We use unbounded input data within our examples, and so we provide the propagation rule for the case of bounded data within the supporting information. Additional information about other propagation rules is available within Samek et al. (2019).
The rules for LRP presented within the literature have thus far been formulated for a specific subset of activation functions, types of neural networks, and neural network tasks. The rules that we present have been developed to work best with the Rectified Linear Unit (ReLU) activation function, since they test whether a node has been “activated” or not (Bach et al., 2015; Montavon et al., 2017). Neurons that use the ReLU activation function are activated in the sense that their output is equal to the input if the input is greater than zero, but is zero if the input is less than zero (see Figure 1b for an illustration of the ReLU function). So, the formulation of LRP that we use ensures that it only traces information back through the network if the nodes are activated and therefore pass information forward when the neural network is making its decision for a particular sample. If the $i$-th node is not activated during the forward pass through the network, then the $a_i$ term is zero in Equation 1, the relevance for the unactivated neuron $i$ is zero, and the relevance is distributed to the other activated neurons within that layer of nodes.

As we have discussed, we use a form of LRP that only propagates information that positively contributes to the output node, which means that the relevance heatmaps show regions that contribute to increases of the output likelihood that a sample belongs to a particular category. This interpretation is helpful for classification tasks, when increasing the likelihood that an input belongs in a particular category is of interest. There are limitations to this approach for regression problems, however, where it is desirable to understand which inputs cause an increase or decrease in the final output. For this reason, we have found that the formulations described by Bach et al. (2015) are not well suited for interpreting neural networks tasked with regression, and we therefore suggest that an LRP formulation needs to be developed specifically for regression problems.

In addition, this formulation of LRP works well for fully-connected neural networks (as we use in this study) and particular formulations of convolutional neural networks. There have been efforts to expand LRP to more complicated neural network architectures, but in these cases other propagation rules need to be used (Arras et al., 2019). It is therefore critical that the neural network architecture be carefully considered prior to training if LRP is to be used.

Additional propagation rules for other cases, such as when negative relevances are to be considered, can be found in the supporting information of this paper or within Montavon et al. (2017) and Samek et al. (2019). We use an implementation of LRP from the authors of the method, which is described in detail within Alber et al. (2019), although an abundance of similar implementations also exist. The implementation we use is available as the `investigate` package within Python, which has been written to work with the Keras neural network package. Tutorials covering how to implement LRP within other programming languages are available at heatmapping.org, and a list of other resources for LRP in Keras and other Python packages is offered within the supporting information.

While there are limitations to LRP, neural networks can be thoughtfully constructed to mediate some of these limitations. For example, many problems of regression can be reformulated as categorical problems by discretizing a continuous output into a number of categories. Additionally, many tasks in geoscience do not seem to require exceedingly complex neural network architectures (e.g. Barnes et al., 2019; Gagne et al., 2019; Ham et al., 2019), and in many cases a basic form of neural network is sufficient to attain high accuracy. Therefore, while the current formulations of LRP do not solve all the limitations of interpreting neural networks for geoscience, we show throughout the remainder of this paper that it still offers opportunities for interpreting neural networks that are thoughtfully constructed with the ultimate objective of interpretation in mind.
4 Applications to Earth System Variability

To illustrate how the interpretation of neural networks can be used to advance scientific knowledge, we apply the optimal input and LRP methods to two well-known patterns of climate variability within the earth system. We intentionally choose patterns that have been extensively researched by the earth system/climate community, because our intent is to demonstrate the usage of neural networks for scientific inference by first showing that the techniques can replicate what we already know before extending into the unknown. Our aim is to provide readers with the intuition and confidence to use the techniques for their own research questions.

For our examples, the inputs to the neural networks are vectorized geospatial fields, the domains of which are discussed in their respective subsections. The neural network is tasked with identifying which of two categories the input geospatial fields are associated with, and the specific categories depends on the example. It is worth noting that optimal input and LRP can be applied to neural networks with any number of output categories, but we limit the output to two categories for the sake of illustration. Additional details about the neural network architectures we use are discussed in Section 2 and the appendix.

4.1 The El Niño-Southern Oscillation (ENSO) Pattern

The first example we use is the simpler of the two, and shows how the optimal input and LRP methods can be used to interpret a neural network’s understanding of the spatial structure of a well-known climate pattern. We show that optimal input is useful for gaining a composite interpretation of the neural network’s understanding of the climate pattern, and that LRP extends beyond this composite and also allows the interpretation of what information is useful to the neural network for each individual sample.

A neural network is tasked with identifying whether a sea-surface temperature (SST) pattern is characteristic of a positive (El Niño) or negative (La Niña) phase of the El Niño Southern Oscillation (ENSO). ENSO is a dominant mode of earth-system variability that acts on an interannual timescale and manifests as sea-surface temperature anomalies within the tropical Pacific, although its indirect influences on weather and climate are global (Philander, 1983; Rasmusson & Wallace, 1983). We use the monthly Niño3.4 index downloaded from https://climatedataguide.ucar.edu/climate-data/ to define the state of ENSO, which is based on an average of sea-surface temperature anomalies within the east-central tropical Pacific (between $5^\circ$S to $5^\circ$N and $170^\circ$W to $120^\circ$W). According to this index, negative sea-surface temperature anomalies within the east-central tropical Pacific are characteristic of La Niña, while positive sea-surface temperature anomalies are characteristic of El Niño. Composite sea-surface temperature anomalies for each phase are shown in Figure 4.

For the neural network setup (shown in Figure 5), the first index of the label vector corresponds to La Niña samples and the second index to El Niño samples. An example vector label for a La Niña case is therefore $[1, 0]$, and the output of the neural network is of similar form with the output value in each index corresponding to the network’s estimated likelihood that the sample belongs in each category. The input dataset is monthly sea-surface temperature anomalies for the years 1880 through 2017 from the 1° by 1° Cobe V2 dataset (Hirahara et al., 2017). We calculate the anomalies separately for each grid point by removing the mean for the years 1980 through 2009 and thereafter removing the linear trend. Samples from the years 1880 through 1990 are used to train the network and those from 1990 through 2017 are used to test the network, and we only test and train on months during which the Niño3.4 index magnitude was greater than 0.5. The network does not see the 1990 through 2017 samples during training, and those samples are only used to test whether what the network learns during training generalizes.
to samples on which the network was not trained. We vectorize the global images of sea-surface temperature anomalies before inputting them into the neural network.

The trained network identifies the ENSO phase with 100% accuracy on both the training (654 samples) and testing (168 samples) datasets. It is uncommon for a neural network to have 100% accuracy, which highlights the intended simplicity of this example. Regardless, in order to achieve this accuracy, the weights and biases of the neural network must contain information about the spatial patterns of sea-surface temperature variability characteristic of ENSO. The high accuracy also implies that the patterns the network uses to identify the phase of ENSO within the training samples are also found within the testing samples.

We focus on the interpretation of the neural network’s understanding of El Niño, although the interpretation for La Niña is similar and provided in the supporting information. We first generate the optimal input to identify the composite spatial pattern of sea-surface temperature anomalies that maximizes the network’s confidence that the sample is an El Niño event (Figure 6a) and the composite relevance heatmaps for all of the El Niño samples (Figure 6b). Then, we use LRP to identify the regions on which the network focuses its attention for El Niño events on a sample-by-sample basis (Figure 7). The relevance values output from LRP for each sample are normalized to range from 0 to 1 by dividing each heatmap by its own maximum relevance value. We do this so that the relevances for each sample are weighted equally when compositing the relevance across samples.

The neural network identifies the canonical signature of El Niño events with remarkable similarity to the observed pattern (Figure 6). The optimal input recovers a map of sea-surface temperature anomalies that is similar to the observed ENSO pattern in both spatial structure and magnitude, particularly within the tropical Pacific (Figure 6a,c). There are some differences in the sign and magnitude of the anomalies outside of the tropical Pacific, such as in the Atlantic Ocean, although these regions are not conventionally considered to be a part of the predominant ENSO pattern (Philander, 1983). The composite relevance for the El Niño samples also shows that the neural network mainly focuses its attention on the tropical Pacific (Figure 6b). A region of non-zero relevance exists within the North Pacific (Figure 6b), which may be associated with a well-known correlation between oceanic variability within this region and the tropical signal of ENSO (Zhang et al., 1996).

The utility of LRP is further highlighted by analyzing relevance heatmaps for individual samples. Figure 7a shows examples of eastern Pacific and central Pacific (i.e. Modoki; Ashok et al., 2007) ENSO events in 1998 and 1987 respectively, and highlights that the network refocuses its attention on different regions of the tropical Pacific to identify an El Niño depending on the input. Furthermore, the neural network focuses its attention on the regions of sea-surface temperature anomalies that are most commonly associated with the two types of El Niño, and learns to ignore other anomalies of similar magnitude within the western Pacific that are distinct from ENSO. We only show the spatial relevance patterns for these two examples, although the relevance time series for the central and eastern Pacific show that the network correctly refocuses its attention for all of the input samples depending on the type of El Niño event (Figure 7c). Samples associated with central Pacific El Niño events have higher relevance within the central Pacific than within the eastern Pacific, and vice versa for samples associated with eastern Pacific El Niño events (Figure 7c).

Because the neural network learns the physical structures of the various modes of ENSO, this lends confidence that optimal input and LRP can be used to better our understanding of other, lesser known patterns of earth system variability. This example also highlights the capability of LRP to study what information a neural network uses in its decision-making process for each individual sample. The earth system rarely behaves ac-
cording to a composite, and so the ability to analyze which aspects of each individual sample are important for the neural network’s associated output is particularly useful for gaining new insights into earth-system variability.

4.2 Seasonal Prediction Using the Ocean

To further illustrate the usefulness of the optimal input and LRP methods, we next extend their usage to a slightly more complex example in which we train a neural network to predict a surface temperature response to sea-surface temperature anomalies months in advance. We focus on seasonal prediction, for which the ocean is a predominant source of atmospheric predictability (Collins, 2002; Doblas-Reyes et al., 2013; Dunstone et al., 2011). Specifically, while it is well known that ENSO is a dominant contributor to atmospheric seasonal predictability (Ropelewski & Halpert, 1986; Wolter et al., 1999), there are other regions of oceanic variability that likely also offer extended atmospheric predictability. One such region is the North Pacific, which can impact surface temperature across North America (Capotondi et al., 2019). We therefore predict continental surface temperature anomalies along the west coast of North America, which is more complicated than predicting the phase of ENSO since the neural network must identify the numerous coincident patterns of sea-surface temperature anomalies across different spatial and temporal scales that can contribute to seasonal temperature predictability.

As shown in Figure 8, we train the neural network to predict the sign (above or below zero) of surface temperature anomalies at a location along the west coast of North America (50°N, 240°E) using maps of sea-surface temperature anomalies within the tropics and Northern Hemisphere (north of 20°S). The chosen location, which is denoted by the red dot in subsequent figures, has previously been shown to have extended surface temperature predictability due to sea-surface temperature forcing on seasonal to annual timescales (e.g. Capotondi et al., 2019; Gershunov, 1998). We input sea-surface temperature anomalies from the 1° by 1° Cobe V2 monthly sea-surface temperature anomaly dataset that is linearly interpolated onto a daily basis (Hirahara et al., 2017), and we use the years 1950 to present day. The corresponding daily surface temperature anomaly labels are gathered from the Berkeley Earth Surface Temperatures (BEST; Rohde et al., 2013) dataset, also spanning from 1950 to present day. For both the sea-surface and continental surface temperatures, we calculate the anomalies separately for each grid point by subtracting the mean values for the years 1980 through 2009 and thereafter removing the linear trend. The training dataset spans from 1950 through 2000 (~18,000 samples), and the testing dataset spans from 2000 through 2018 (~7,000 samples). The surface temperature anomalies are averaged over a 60-day period to ensure the predictions are capturing longer-term surface temperature variability, and the averages are centered such that a 60-day prediction implies a prediction of the average 30- to 90-day surface temperature anomalies.

We use interpretations of the neural network to identify which sea-surface temperature patterns are useful for making extended surface temperature predictions at various prediction lead times. We first train a neural network to predict the sign of 60-day averaged surface temperature anomalies 60 days in advance, for which the network has 67% accuracy. We then focus on interpreting the neural network for cases when the surface temperature anomalies are positive, although the interpretation for the cases with negative anomalies is similar and provided within the supporting information. For this lead time, the optimal input and LRP composite identify similar regions of SST patterns that lend predictability (Figure 9a,b). When compared to the composited, correctly classified samples, the optimal input and LRP composites emphasize the tropical Pacific less and highlight a more local forcing within the North Pacific (Figure 9a,b,c). Both of these regions have been identified as sources of predictability on seasonal timescales for the west coast of North America (Capotondi et al., 2019; Gershunov, 1998; Wolter et al., 1999).
We next test the fidelity of the neural network interpretations by varying the prediction lead time of the continental surface temperature anomalies from 180 days prior to 60 days following their occurrence. We compare the neural network interpretations with that of linear regression to both test whether the interpretations are reliable and if they offer any unique insight compared to more conventional approaches. For our linear regression approach, we first obtain a map of regression coefficients by regressing the time series of global maps of sea-surface temperature anomalies onto the time series of surface temperature anomalies over the west coast of North America. We then project this map of regression coefficients onto the observed sea-surface temperature anomalies to predict the sign of the surface temperature at the location along the west coast of North America. The resulting accuracies of the prediction methods and the associated sea-surface temperature patterns are shown in Figure 10.

At extended leads, the spatial patterns of sea-surface temperature anomalies identified by optimal input and LRP are similar to those identified by regression (Figure 10). Particularly, the tropical Pacific stands out as being a predominant source of surface temperature predictability across the 180-day, 120-day, and 60-day prediction lead times for both the neural network interpretation and the regression maps (Figure 10a,b,c). However, for the 60-day prediction lead time, within the neural network interpretation the importance of the ENSO region within the tropical Pacific diminishes relative to the North Pacific anomalies, and further lessens for the concurrent and 60-day lagged sea-surface temperature anomalies (Figure 10c,d,e). Unlike the neural network, the regression approach continues to highlight the tropical Pacific Ocean as important for the concurrent and 60-day lagged sea-surface temperature anomalies.

The neural network is more accurate than the regression approach for all prediction ranges, suggesting that the spatial patterns identified by the interpreted network are likely more physically representative of the patterns that force the surface temperature response. Specifically, the neural network interpretations suggest that the North Pacific is a predominant modulator of concurrent surface temperature anomalies along the west coast of North America, while the tropical Pacific offers extended lead predictability. This idea is corroborated by previous research that found the North Pacific to modulate temperatures across western North America separately from the tropical Pacific (Capotondi et al., 2019).

5 Discussion and Conclusions

The recent surge in the popularity of neural networks within the geosciences has inspired the need for techniques to interpret their decisions. Neural networks are conventionally thought of as “black boxes” within the geosciences with limited tools for the interpretation of the reasoning behind their decision-making process. We have shown that the usage of two separate techniques enables physically meaningful inference from thoughtfully designed neural networks. This ability to reliably interpret these types of neural networks opens the door to extending upon a network’s output by using the interpretation of how and why the network makes its decisions as the ultimate science outcome.

The optimal input method can be used to quantify the patterns within the input data that maximize a neural network’s confidence that an input is associated with a particular output. For the case of categorical output as we present within this paper, optimal input iteratively changes an input to maximize the neural network’s confidence that it belongs in a particular category. The optimized input has the same dimensions and can be interpreted in the same units as the input samples used to train the network, but provides no direct indication as to which regions within the image are most important for a specific output.
Layerwise relevance propagation (LRP), on the other hand, considers each sample individually, and provides information about which features in each sample are most important, or relevant, for the network’s associated output. LRP can thereby provide insight into how relationships between the inputs and outputs of a neural network vary on a case-by-case basis. The usefulness of this quality is exemplified by comparing the relevance heatmaps for two types of El Niño events – the eastern Pacific and central Pacific, or Modoki, patterns (Figure 7). Although the optimal input pattern does not distinguish between these two modes of El Niño variability because it offers a composite interpretation (Figure 6a), LRP shows that the network does redirect its focus depending on where the sea-surface temperature anomalies occur (Figure 7). The fact that the neural network learns the variable spatial structures of ENSO, and that LRP can elucidate this understanding, suggests that LRP can be used to identify physically meaningful patterns within other geoscientific datasets, as well.

There are, however, particular requirements of the optimal input and LRP techniques that constrain how a neural network is constructed, the details of which are discussed in Section 3. We therefore emphasize that neural networks must be constructed thoughtfully so as to maximize the scientific value of their interpretation. The network architecture must be complex enough to capture any existing relationships between the input and output data, but not so complex such that interpretation methods are no longer usable. The relative value of the accuracy and interpretability of a neural network is of critical importance to scientific analyses, and should be assessed carefully prior to training. If a network is too simple to accurately capture the relationships between the input and output, then its accuracy will be low and any interpretations of its understanding will be limited in scientific value. On the other hand, if a network is too complex and interpretation is impossible, then the network is limited to solely its output. A balance between network complexity and interpretability must be struck if the interpretation of what a network has learned is to be scientifically useful.

We have shown that techniques for interpreting neural networks have the potential to extend the usage of neural networks to the discovery of unknown patterns within geoscientific data. The ultimate scientific outcome of a neural network can now also be the interpretation of what the neural network has learned, rather than just the output of the network itself. Regardless of the specific application, it is now apparent that neural networks offer scientists a powerful new way to identify and understand undiscovered connections within geoscience data.
Neural Network Details

a) Neural Network Architecture

b) ReLU Activation Function

\[ f(x) = \max(0, x) \]

c) Example Node

\[ a_{2,2} = \max \left( 0, \sum_{i} a_{1,i}w_{1,i;2,2} + b_{2,2} \right) \]

Figure 1. Illustration of the neural network architecture used here.
**Illustration of the Optimal Input Procedure**

1) Train network & freeze weights/biases
   - hidden layer
   - frozen weights/biases
   - input layer
   - output layer

   information learned during training:
   - positive weights/biases
   - negative weights/biases

2) Identify desired output
   - 1
   - 0

3) Generate all-zero input
   - 0
   - 0
   - 0

4) Pass forward the all-zero input and calculate loss
   - network output
   - desired output
   - Loss = \(- \sum_i y_i \log(\hat{y}_i)\)
   - Loss = \(- 1 \times \log(0.1) - 0 \times \log(0.9)\)
   - Loss = 2.30

5) Backpropagate loss through frozen weights to update input
   - Loss = 2.30

6) Iterate through steps 4 and 5 until the loss converges...
   - pass sample forward
   - Loss = 1.61
   - Loss = 0

**Figure 2.** Illustration of the optimal input procedure used here for interpreting neural networks. The steps illustrated here correspond to the steps listed in Section 3.1. The neural network within this schematic has already been trained, and the training procedure is not illustrated.
Illustration of Layerwise Relevance Propagation

1) Train network & freeze weights/biases
   - hidden layer
   - input layer
   - output layer
   - frozen weights/biases
   - Information learned during training:
     - positive weights/biases
     - negative weights/biases

2) Input sample into frozen network and retain output
   - full output
   - selected output node to propagate relevance

3) Propagate relevance from output node to previous layer
   - propagation rule
   \[ R_i = \sum_j \left( \frac{w_{ij}^2}{\sum_i w_{ij}^2} \right) R_j \]
   - example relevance calculations
     \[ R_{i=1} = \left( \frac{a_1 w_{1:2}}{a_1 w_{1:2} + a_2 w_{2:2}} \right) \hat{y}_2 \]
     \[ R_{i=2} = \left( \frac{a_2 w_{2:1}}{a_1 w_{1:2} + a_2 w_{2:2}} \right) \hat{y}_2 \]

4) Propagate relevance from hidden layer to input layer
   - propagation rule
   \[ R_i = \sum_j \left( \frac{w_{ij}^2}{\sum_i w_{ij}^2} \right) R_j \]
   - all biases are ignored for this rule
   - relevance at input layer
     \[ R_1, R_2, R_3 \]

5) Repeat for each sample of interest...

**Figure 3.** Illustration of the layerwise relevance propagation (LRP) procedure used here for interpreting neural networks. The steps illustrated here correspond to the steps listed in Section 3.2. The neural network within this schematic has already been trained, and the training procedure is not illustrated. While the illustration does not show the propagation from one hidden layer to another hidden layer, the propagation method is identical to the propagation from the output node to the final hidden layer.
Figure 4. Composites of the monthly sea-surface temperature anomalies during (a) El Niño (337 samples) and (b) La Niña (485 samples). The composites have been normalized to have a maximum absolute sea-surface temperature anomaly of 1 and include all events with a Niño3.4 index magnitude of greater than 0.5.
Neural Network Design for ENSO Phase Identification

Figure 5. Illustration of the neural network design for ENSO phase identification.
Figure 6. Interpretation of the neural network’s understanding of the spatial structure of El Niño based on 337 total El Niño samples (including both training and testing data). a) The optimal input field that shows the input image that maximizes the confidence of the network that the sample is an El Niño event. b) The LRP composite for all El Niño events, where higher values denote greater relevance for the network’s decision. Relevance values are normalized between 0 and 1 for each sample, such that 1 denotes the highest relevance in each individual sample and 0 denotes the lowest relevance. c) Composite observed monthly sea-surface temperature anomalies for all El Niño samples (Niño3.4 > 0.5), identical to what is shown in Figure 4.
Figure 7. An illustration of how the neural network focuses on different regions of sea-surface temperature anomalies for different types of El Niño: a) an eastern Pacific El Niño event and b) a central Pacific (Modoki) El Niño event. The observed sea-surface temperature anomalies for each case are shown in fill and the LRP relevance is contoured. The relevance has been normalized to lie on a scale from 0 to 1, and the contours range in value from 0.2 to 1.0 in increments of 0.2. Relevance values less than 0.2 have been omitted. c) (top) The Niño3.4 index time series from 1968 to 2011; (bottom) Time series of the normalized relevance values for locations within the central Pacific and eastern Pacific from 1968 through 2011. Relevance values are only shown for months during which the Niño3.4 index was greater than 0.5. The central (eastern) Pacific location is denoted by the orange (purple) dot in panels a (b), and is located on the equator at a longitude of 200° (250°). The types of each El Niño event during the 1968 through 2011 period are as noted in Ashok et al. (2007), Lee and McPhaden (2010), and Wang and Wang (2014), and are denoted above the time series as either central (“C”) or eastern Pacific (“E”) events. If an event was not determined to be separable into a central or eastern Pacific event by Ashok et al. (2007), Lee and McPhaden (2010), or Wang and Wang (2014), then it is not labeled.
Neural Network Design for Seasonal Prediction Example

Figure 8. Illustration of the neural network design for the seasonal prediction example.

Each input node = SST at one grid point

Hidden Layers (2 layers, 8 nodes each)

Output Layer (2 nodes)

likelihood that temperature anomaly at red dot < 0°C

likelihood that temperature anomaly at red dot > 0°C

\[ y_1 \]

\[ y_2 \]
Figure 9. Interpretation of the neural network’s understanding of the sea-surface temperature anomalies that lend predictability for positive surface temperature anomalies at the red dot 60 days in advance based on $\sim 12,000$ total samples (including both training and testing data). a) The optimal input field that shows the input image that maximizes the confidence of the network that the input sea-surface temperature samples is associated with positive temperature anomalies at the red dot. b) The LRP composite for all correctly categorized samples of positive temperature anomalies, where higher values denote greater relevance for the network’s decision. Relevance values are normalized between 0 and 1 for each sample, such that 1 denotes the highest relevance in each individual sample and 0 denotes the lowest relevance. c) Composite observed sea-surface temperature anomalies for all cases where the neural network accurately predicted positive surface temperature anomalies.
Figure 10. A comparison of the spatial patterns of sea-surface temperature deemed important for predicting surface temperature at the red dot using a neural network and linear regression. An evolution of the sea-surface temperature patterns at various lead times are shown, ranging from 180 days prior to the surface temperature anomalies to 60 days after. For each lead/lag, the top panel shows the neural network optimal input in fill and LRP relevance in open contours, and the bottom panel shows the regression coefficients for the linear regression approach. The open contours denote relevance values ranging from 0.1 to 0.3 in increments of 0.05.
Appendix A Additional Neural Network Details

The individual grid cells within the vectorized inputs, which are maps in our cases, are each treated as independent inputs of the neural network. Each input node receives the value for one element of the input vector and is connected to each node within the first hidden layer of internal nodes. The individual nodes of the first hidden layer are therefore each connected to every element of the input vector, and can use information from any input element according to the weight connecting the node to the inputs. The first hidden layer is then connected to the second hidden layer in a similar fashion, with each node within the first hidden layer connected to each node within the second hidden layer. The Rectified Linear Unit (ReLU) activation is applied to the output from each of the hidden layer nodes. Each node within the second hidden layer is finally connected to the two output nodes, which represent the neural network’s estimated likelihood that the input sample corresponds to each of the two categories. The weights and biases are initialized randomly using the “He normal” technique (He et al., 2015), such that they do not contain any information about the relationship between the inputs and outputs upon initialization. When the neural network is trained, the weights and biases of the network are iteratively updated such that the output of the network is most similar to the input labels (i.e. the model is most accurate) once the network’s weights and biases have converged on an optimal solution.

The likelihood output is generated by applying a “softmax” operator to the output of the neural network before estimating its accuracy, which is formulated as follows:

\[
\tilde{y}_i = \frac{\exp(x_i)}{\sum_j \exp(x_j)}
\] (A1)

where \(x_i\) represents the pre-softmax output of the neural network for output node \(i\) (of which there are two in our formulation), the numerator is the exponential of the value of that output node, and the denominator is the sum of the exponential of all pre-softmax outputs. In this sense, the post-softmax output of the neural network is a relative likelihood that the input sample belongs to each class, with higher values being indicative of a higher likelihood, and vice versa. Following the application of the softmax operator, we then use the cross-entropy loss function to estimate the accuracy of the network, which takes the form of:

\[
\text{loss} = -\sum_i y_i \log(\tilde{y}_i)
\] (A2)

where \(i\) represents the \(i\)-th unit of the label vector for the input sample, \(y_i\) is the value of the \(i\)-th unit of the label vector, and \(\tilde{y}_i\) is the output value of the \(i\)-th node of the output layer from the neural network after being transformed by the softmax operator. This loss function therefore assigns error to the output of the neural network on a logarithmic scale based on how different the output likelihood vector is from the label of the input sample, and punishes large errors more severely than small errors due to the logarithmic transformation.

The neural networks are trained using gradient descent with the Nesterov accelerated stochastic gradient descent optimizer (Nesterov, 1983; Ruder, 2016). The learning rate is set to an initial value of 0.01 with a Nesterov momentum parameter of 0.9. The learning rate is reduced by a factor of 0.5 after 50 epochs, and the neural networks are trained for a total of 100 epochs, which is sufficient for convergence for all of the examples within this paper.

We use L2 (i.e. ridge) regularization for each example to ensure the network divides its attention across a greater number of input nodes than it otherwise would. For the ENSO problem, we use an L2 parameter of 25 for the weights between the input layer and the first hidden layer and 0.01 for all other weights. For the seasonal prediction problem, we use an L2 parameter of 10 for the weights between the input layer and the first
hidden layer and 0.01 for all other weights. We find that a careful selection of the $L_2$ parameter is important for ensuring that the neural network does not overfit to the input data, although our conclusions are consistent for $L_2$ parameters of 5 to 50 between the input layer and first hidden layer.

A more extended review of neural networks and their various forms are available through other resources (e.g. Gagne et al., 2019; Gers et al., 1999; Simon, 1994).

Acknowledgments
Benjamin A. Toms was funded by the Department of Energy Computational Science Graduate Fellowship via grant DE-FG02-97ER25308. Elizabeth A. Barnes was funded, in part, by NSF-AGS CAREER grant AGS-1749261. Support was also provided to Imme Ebert-Uphoff and Elizabeth A. Barnes through NSF grant 1934668 of the HDR (Harnessing the Data Revolution) program.
References

Alber, M., Lapuschkin, S., Seegerer, P., Hägele, M., Schütt, K. T., Montavon, G., ... Kindermans, P.-J. (2019). iNNvestigate Neural Networks! Journal of Machine Learning Research, 20(93), 1-8.

Arras, L., Arjona-Medina, J., Widrich, M., Montavon, G., Gillhofer, M., Müller, K.-R., ... Samek, W. (2019). Explaining and interpreting LSTMs. In Explainable ai: Interpreting, explaining and visualizing deep learning (pp. 211–238). Springer.

Ashok, K., Behera, S. K., Rao, S. A., Weng, H., & Yamagata, T. (2007). El Niño Modoki and its possible teleconnection. Journal of Geophysical Research: Oceans, 112(C11).

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one, 10(7), e0130140.

Barnes, E. A., Hurrell, J. W., Ebert-Uphoff, I., Anderson, C., & Anderson, D. (n.d.). Viewing forced climate patterns through an AI Lens. Geophysical Research Letters. doi: 10.1029/2019GL084944

Bergen, K. J., Johnson, P. A., Maarten, V., & Beroza, G. C. (2019). Machine learning for data-driven discovery in solid earth geoscience. Science, 363(6433), eaau0323.

Bolton, T., & Zanna, L. (2019). Applications of deep learning to ocean data inference and subgrid parameterization. Journal of Advances in Modeling Earth Systems, 11(1), 376–399.

Boukabara, S.-A., Krasnopolsky, V., Stewart, J. Q., Penny, S. G., Hoffman, R. N., & Maddy, E. (2019, August). Artificial intelligence may be key to better weather forecasts. EOS.

Brenowitz, N. D., & Bretherton, C. S. (2018). Prognostic validation of a neural network unified physics parameterization. Geophysical Research Letters, 45(12), 6289–6298.

Capotondi, A., Sardeshmukh, P. D., Di Lorenzo, E., Subramanian, A. C., & Miller, A. J. (2019). Predictability of US West Coast Ocean Temperatures is not solely due to ENSO. Scientific reports, 9(1), 10993.

Collins, M. (2002). Climate predictability on interannual to decadal time scales: The initial value problem. Climate Dynamics, 19(8), 671–692.

Doblas-Reyes, F. J., García-Serrano, J., Lienert, F., Biescas, A. P., & Rodrigues, L. R. (2013). Seasonal climate predictability and forecasting: Status and prospects. Wiley Interdisciplinary Reviews: Climate Change, 4(4), 245–268.

Dunstone, N., Smith, D., & Eade, R. (2011). Multi-year predictability of the tropical Atlantic atmosphere driven by the high latitude North Atlantic Ocean. Geophysical Research Letters, 38(14).

Feng, X., Li, Q., Zhu, Y., Hou, J., Jin, L., & Wang, J. (2015). Artificial neural networks forecasting of PM2.5 pollution using air mass trajectory based geographic model and wavelet transformation. Atmospheric Environment, 107, 118–128.

Gagne, D. J., Haupt, S. E., Nychka, D. W., & Thompson, G. (2019). Interpretable deep learning for spatial analysis of severe hailstorms. Monthly Weather Review, 147(8), 2827–2845.

Gardner, M., & Dorling, S. (1999). Neural network modelling and prediction of hourly NOx and NO2 concentrations in urban air in London. Atmospheric Environment, 33(5), 709–719.

Gers, F. A., Schmidhuber, J., & Cummins, F. (1999). Learning to forget: Continual prediction with LSTM.

Gershunov, A. (1998). ENSO influence on intraseasonal extreme rainfall and temperature frequencies in the contiguous United States: Implications for long-range predictability. Journal of Climate, 11(12), 3192–3203.
Gil, Y., Pierce, S. A., Babaie, H., Banerjee, A., Borne, K., Bust, G., ... others (2018). Intelligent systems for geosciences: an essential research agenda. Communications of the ACM, 62(1), 76–84.

Ham, Y.-G., Kim, J.-H., & Luo, J.-J. (2019). Deep learning for multi-year ENSO forecasts. Nature, 1–5.

Haykin, S. (1994). Neural networks: A comprehensive foundation. Prentice Hall PTR.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the ieee international conference on computer vision (pp. 1026–1034).

Hirahara, S., Ishii, M., & Fukuda, Y. (2017). Centennial-scale sea surface temperature analysis and its uncertainty. Journal of Climate, 30(20).

Karpatne, A., Ebert-Uphoff, I., Ravela, S., Babaie, H. A., & Kumar, V. (2018). Machine learning for the geosciences: Challenges and opportunities. IEEE Transactions on Knowledge and Data Engineering.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097–1105).

Lagerquist, R., McGovern, A., & Gagne, D. J. (2019). Deep learning for spatially explicit prediction of synoptic-scale fronts. Weather and Forecasting, 34(4), 1137–1160.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436.

Lee, T., & McPhaden, M. J. (2010). Increasing intensity of El Niño in the central-equatorial Pacific. Geophysical Research Letters, 37(14).

Lopatka, A. (2019, May). Meteorologists predict better weather forecasting with AI. Physics Today, 72(5), 32–34.

Malde, K., Handegard, N. O., Eikvil, L., & Salberg, A.-B. (2019). Machine intelligence and the data-driven future of marine science. ICES Journal of Marine Science.

McGovern, A., Elmore, K. L., Gagne, D. J., Haupt, S. E., Karstens, C. D., Lagerquist, R., ... Williams, J. K. (2017). Using artificial intelligence to improve real-time decision-making for high-impact weather. Bulletin of the American Meteorological Society, 98(10), 2073–2090.

McGovern, A., Lagerquist, R., Gagne, D. J., Jergensen, G. E., Elmore, K. L., Homeyer, C. R., & Smith, T. (2019). Making the black box more transparent: Understanding the physical implications of machine learning. Bulletin of the American Meteorological Society(2019).

Montavon, G., Lapuschkin, S., Binder, A., Samek, W., & Müller, K.-R. (2017). Explaining nonlinear classification decisions with deep taylor decomposition. Pattern Recognition, 65, 211–222.

Nesterov, Y. (1983). A method for unconstrained convex minimization problem with the rate of convergence O (1/k²). In Doklady an ussr (Vol. 269, pp. 543–547).

O’Gorman, P. A., & Dwyer, J. G. (2018). Using machine learning to parameterize moist convection: Potential for modeling of climate, climate change, and extreme events. Journal of Advances in Modeling Earth Systems, 10(10), 2548–2563.

Olah, C., Mordvintsev, A., & Schubert, L. (2017). Feature visualization. Distill, 2(11), e7.

Philander, S. G. H. (1983). El Nino southern oscillation phenomena. Nature, 302(5906), 295.

Rasmussen, E. M., & Wallace, J. M. (1983). Meteorological aspects of the El Nino/southern oscillation. Science, 222(4629), 1195–1202.

Rasp, S., Pritchard, M. S., & Gentine, P. (2018). Deep learning to represent subgrid processes in climate models. Proceedings of the National Academy of Sciences,
115(39), 9684–9689.
Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., et al. (2019). Deep learning and process understanding for data-driven earth system science. *Nature*, 566(7743), 195.
Rohde, R., Muller, R., Jacobsen, R., Muller, E., Perlmutter, S., Rosenfeld, A., . . . Wickham, C. (2013). A new estimate of the average earth surface land temperature spanning 1753 to 2011. *Geoinformatics and Geostatistics: An Overview*, 7, 2.
Rolnick, D., Donti, P. L., Kaack, L. H., Kochanski, K., Lacoste, A., Sankaran, K., . . . others (2019). Tackling climate change with machine learning. *arXiv preprint arXiv:1906.05433*.
Ropelewski, C. F., & Halpert, M. S. (1986). North American precipitation and temperature patterns associated with the El Niño/Southern Oscillation (ENSO). *Monthly Weather Review*, 114(12), 2352–2362.
Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.
Samek, W. (2019). *Explainable AI: Interpreting, explaining and visualizing deep learning*. Springer Nature.
Simonyan, K., Vedaldi, A., & Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.
Toms, B. A., Kashinath, K., Yang, D., et al. (2019). Deep learning for scientific inference from geophysical data: The Madden-Julian oscillation as a test case. *arXiv preprint arXiv:1902.04621*.
Wang, X., & Wang, C. (2014). Different impacts of various El Niño events on the Indian Ocean Dipole. *Climate dynamics*, 42(3-4), 991–1005.
Wolter, K., Dole, R. M., & Smith, C. A. (1999). Short-term climate extremes over the continental United States and ENSO. Part I: Seasonal temperatures. *Journal of Climate*, 12(11), 3255–3272.
Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., & Lipson, H. (2015). Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.0579*. 
Zhang, Y., Wallace, J. M., & Iwasaka, N. (1996). Is climate variability over the North Pacific a linear response to ENSO? *Journal of Climate*, 9(7), 1468–1478.