GAN for Recommendation System

Prosvetov A. V. 1, 2

1 CleverDATA
2 Space Research Institute Russian Academy of Science

Abstract. In our day there are many approaches for recommendation generation, however several unsolved problems still exist. In our work we study the potential of Generative Adversarial Networks to generate relevant recommendations and compare the proposed approach with another recommendations system based on neural networks. We found that recommendation system based on GANs can successively compete with state-of-the-art network in the field of recommendation generation and suggest, that GAN recommendation system have high potential.

1 Introduction

Nowadays, the problem of recommendation generation can be solved in several ways. One of the methods is based on matrix decomposition (Collaborative Filtering, Latent Factor analysis, ALS and etc). Another family of approaches is based on neural networks (auto-encoders, Deep Semantic Similarity Model, Restricted Boltzmann machine, etc.). Neural network approaches allow to use additional information about products and sequence of actions in embeddings [1], however tensor approaches are still widely used and show good results. Despite the successes achieved, there are several unresolved issues in the field of recommendation systems.

For example, a recommendation system can give each client ratings for each product, but the ratings obtained are fixed numbers, while in some cases distribution is required. Another problem is that after proposing a recommendations one can obtain positive and negative outcome, thus the reinforced learning approach is more suitable.

Algorithms that try to learn p(y|x) directly or algorithms that try to learn mappings from space of inputs X to the labels \{0,1\} are discriminative. In opposite way the algorithms that try to learn p(x|y) are generative. For instance, if y indicates whether a product is attractive to customer, then p(x|y = 1) models the distribution of relevant product features, and p(x|y = 0) models the distribution of non-relevant product features for the customer. In our study, we are trying to solve the problems of recommendation systems using generative approach called the Generative Adversarial Network (GAN), proposed in 2014 by Ian Goodfellow and co-authors [2]. The GAN type of neural networks consists of two parts: the first tries to generate new samples with the same statistical characteristics as the training set, and the second network learns to distinguish between real and synthetic samples. After the training
process, one can get a generative model for the training set, which can generate new samples similar to the training data. Originally proposed for unsupervised learning, the GAN approach can be applied in reinforced learning [3], which gives high potential for its applications in the generation of recommendations.

Typically, GANs are used for image generation and photorealistic results were achieved in recent works [4]. Recently GANs were used to model the distribution of dark matter in a particular direction in space and to predict the gravitational lensing that will occur [5]. GANs have also been proposed as a fast and accurate way of generating simulated showers of particles in the calorimeters of high-energy physics experiments [6,7]. However, the generalization potential of GANs can be used in the field of recommendation systems to propose more relevant offers.

2 GAN details

Let’s assume models of Generator and Discrimination are multilayer perceptrons. Let’s define a prior on input noise variables $p_{\theta}(z)$ to learn the generator’s distribution $p_g$ over data $x$. The Generator represent a mapping to data space as $G(z; \theta_g)$, where $G$ is a differentiable function represented by a multilayer perceptron with parameters $\theta_g$. Let’s define Discriminator as second multilayer perceptron $D(x; \theta_d)$ that outputs a single scalar. We can generate train set where $D(x)$ represents the probability that $x$ came from the data rather than form the Generator. Next we can train $D$ to maximize the probability of assigning the correct label to both training examples and samples from $G$. We can train simultaneously $G$ to minimize $\log(1-D(G(z)))$. Thus, $D$ and $G$ play the following two-player minimax game with value function $V(G, D)$:

$$
\min_{G} \max_{D} V(G, D) = E_{x \sim p_{data}}[\log D(x)] + E_{z \sim p_{z}}[\log (1 - D(G(z)))]
$$

To avoid overfitting one can alternate between $k$ steps of optimizing $D$ and one step of optimizing $G$, that results in $D$ being maintained near its optimal solution, so long as $G$ changes slowly enough.

Discriminator and Generator can be deep neural networks with LSTM recurrent cells [8] and convolutional layers [9] as we used in our work.

Generative adversarial nets can be extended to a conditional model if both the generator and discriminator are conditioned on information $y$ [10], in our case we used information about user’s historical choices. The conditioning was performed by feeding into both the discriminator and generator an additional input layer.

The objective function of a two-player minimax game would be as Eq (2):

$$
\min_{G} \max_{D} V(D, G) = E_{x \sim p_{data}}[\log D(x|y)] + E_{z \sim p_{z}}[\log (1 - D(G(z)|y))]
$$

Fig 1 illustrates the structure of simple conditional adversarial network for modified for recommendation generation.
3 Recommendation system

The GAN recommendation system architecture consists of two neural networks and an additional intermediate layer.

1) The first neural network is a generator, and the goal of this network is to create a new user choice based on historical choices and noise. The generator has two inputs: the historical choice of the client and additional noise. The embeddings for historical choices are trainable. The Generator output is a fully connected layer with a sigmoid activation function, the size of the layer is equal to the number of potential customer choices options.

2) The second neural network is a discriminator, and the purpose of this network is to determine if the input is a valid or synthetic example. The discriminator also has two inputs: the historical choice of the client and the subsequent choice. The embeddings for historical choice are trainable. The discriminator output is a single neuron with a sigmoid activation function.

3) An additional level is the level of Gaussian noise, and it is necessary in order to avoid rounding operations that do not allow backpropagation learning. The Gaussian noise level is located between the generator and the discriminator and has a size equal to the number of potential customer choices options. The density for the Gaussian distribution is the following:

\[ p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right) \]  \hspace{1cm} (3)
where \( \Sigma \) is a covariance matrix \( (\Sigma \in \mathbb{R}^{n\times n}) \) and \( \mu \) is a mean vector \( (\mu \in \mathbb{R}^n) \).

The Gaussian noise level was used only on stage of model training and was ignored on stage of inference.

As can be seen from the description of the architecture, the following information is required for the training set:
- historical customer choice;
- next customer choice.

Input noise for the first network is randomly generated during training. Please note that we do not need to generate a synthetic negative choice for the client, because the cases generated by the Generator will play the role of negative objects for the Discriminator. The generation of negative choices can cause a number of problems in the learning process, because the neural network is trying to use the frequency of choice as additional information. Our approach avoids the complications with synthetic negative samples.

4 RESULTS

To compare new approach with existing State-of-the-Art model we prepared the recommendations for airline tickets and tested them using personal offers sent by email.

The architecture of Generator was encoder-decoder type of network: encoder had 6 convolution layers with relu activation function and batch normalization, decoder had 3 convolution layers with relu activation function, upsampling and batch normalization.

The architecture of discriminator was recurrent type of neural network: we used 3 LSTM layers and 3 fully connected layers with dropout and batch normalization.

The DSSM network had 6 convolution layers with relu activation function, max pooling and 3 fully connected layers with dropout and batch normalization.

The comparison of GAN recommender with DSSM [11] neural network is presented in table 1. We tested recommendations on 162,978 clients using A/B testing approach with control group.
Table 1. The comparison of recommendation systems.

|                       | DSSM   | GAN    | Baseline |
|-----------------------|--------|--------|----------|
| Total communicated clients | 69,688 | 59,602 | 33,688   |
| Total conversions     | 1,627  | 1,393  | 744      |
| Conversion Rate       | 2.33 +/- 0.11 % | 2.34 +/- 0.12 % | 2.21 +/- 0.16 % |
| Relative Uplift       | 5.71%  | 5.83%  | -        |

As one can see GAN recommendations can successively compete with DSSM network in the field of airline tickets. The difference between DSSM and GAN results is not significant, however, the customers received relevant offers from both networks and conversion rate is higher than the baseline in both cases. The DSSM network was well designed and prepared for the domain area, thus GAN network was compared to a strong competitor. The further improvement of GAN recommendations network can be achieved using additional domain information.

5 Conclusion

In our work we study the potential of Generative Adversarial Networks in generation of relevant recommendations. The competitive advantage of GAN recommender is that no negative objects require in training set. Also, one can obtain for each user a distribution of rating for each choice. The proposed architecture can meet problems in cases, where user has a large number of options. The recommendation system based on GAN neural network was trained and successively competed with state-of-the-art recommendation system based on DSSM architecture. The further progress in GAN recommendation networks can be achieved using additional domain information, season trends and etc.
References

1. A Prosvetov, “The comparison of autoencoder architectures in improving of prediction models”, Journal of Physics: Conference Series, Volume 1117, conference 1 (2018)

2. Goodfellow, Ian; Pouget-Abadie, Jean; Mirza, Mehdi; Xu, Bing; Warde-Farley, David; Ozair, Sherjil; Courville, Aaron; Bengio, Yoshua. “Generative Adversarial Networks”. Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014). pp. 2672–2680 (2014)

3. Ho, Jonathon; Ermon, Stefano. "Generative Adversarial Imitation Learning". Advances in Neural Information Processing Systems: 4565–4573 (2016)

4. Horev, Rani. "Style-based GANs – Generating and Tuning Realistic Artificial Faces". Lyrn.AI. (2019)

5. Mustafa, Mustafa; Bard, Deborah; Bhimji, Wahid; Lukić, Zarija; Al-Rfou, Rami; Kratochvil, Jan M. "CosmoGAN: creating high-fidelity weak lensing convergence maps using Generative Adversarial Networks". Computational Astrophysics and Cosmology Simulations, Data Analysis and Algorithms 6:1 (2019)

6. Erdmann, Martin; Glombitza, Jonas; Quast, Thorben. "Precise Simulation of Electromagnetic Calorimeter Showers Using a Wasserstein Generative Adversarial Network". Computing and Software for Big Science. 3: 4 (2019)

7. Musella, Pasquale; Pandolfi, Francesco. "Fast and Accurate Simulation of Particle Detectors Using Generative Adversarial Networks". Computing and Software for Big Science. 2: 8. (2018)

8. S Hochreiter, J Schmidhuber, “Long short-term memory”, Neural computation 9.8: 1735-1780 (1997)

9. Zhang, Wei. "Shift-invariant pattern recognition neural network and its optical architecture". Proceedings of Annual Conference of the Japan Society of Applied Physics (1988)

10. Mehdi Mirza, Simon Osindero. “Conditional Generative Adversarial Nets”. arXiv:1411.1784 (2014)

11. Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, Larry Heck, “Learning Deep Structured Semantic Models for Web Search using Clickthrough Data”, Proceedings of the 22nd ACM international conference on Conference on information & knowledge management (2013)