NEURAL TENSOR NETWORK

IMPLEMENTATION AND APPLICATIONS

Bogdan Kirillov and Anton Dmitriev

TASK STATEMENT

1. Implement neural tensor layer
2. Apply Tucker decomposition to reduce space
3. Add orthogonality constraint support
4. Try to apply it to different tasks

NEURAL TENSOR LAYER ARCHITECTURE

- Neural tensor layer (NTL) is a generalization of dense layer
- In dense layer we use tensors of order 2 and in tensor layer we can use tensor of any arbitrary order
- This approach makes tensor layer of order greater than 2 nonlinear by itself

The output of tensor layer could be written as:

$$Y(X) = b + W_1 \times_1 X + W_2 \times_1 X \times_2 X + ... + W_n \times_1 X ... \times_n X$$

$$X \in R^{N \times I} \quad Y(X) \in R^{N \times O}$$

TENSOR DECOMPOSITION APPROACH

- In this work, we applied tensor decomposition for reducing number of trainable parameters
- Moreover, It could be interpreted as a regularization method
- The idea is pretty straightforward: initialize tensor in Kruskal form and form a full tensor on the each forward step
- We tried to avoid computing the full tensor but it was too slow

$$\begin{cases} W_1 = [[W_{1,1}, W_{1,2}]] \in R^{I \times O} \\ W_2 = [[W_{2,1}, W_{2,2}, W_{2,3}]] \in R^{I \times I \times O} \\ \dots \\ W_n = [[W_{n,1}, W_{n,2}, ..., W_{n,n+1}]] \in R^{I \times I \times \dots \times O} \end{cases}$$

$$\begin{cases} W_{i,j} \in R^{I \times R} \text{ where } R \text{ is decomposition rank if } j < i+1 \\ W_{i,i+1} \in R^{O \times R} \end{cases}$$

OTHOGONALITY CONSTRAINT

WE USED MODIFIED LOSS FUNCTION TO ADD ORTHOGONALITY CONSTRAINT

$$Finalloss(X, Y_{true}) = loss(Y(X), Y_{true}) + \sum_{i=1}^{n} \sum_{j=1}^{i+1} sum\left(\left(W_{i,j}^{T} W_{i,j} - I_R\right)^2\right)$$

In our experiments, Base loss is Binary Crossentropy (BCELoss)

APPLY METHOD TO THE DATA

We choose two methods to apply our model

1. Kaggle diabetes classification dataset (https://www.kaggle.com/johndasilva/diabetes)
2. Mechanisms of Action Prediction competition on kaggle (https://www.kaggle.com/c/lish-moa/overview)

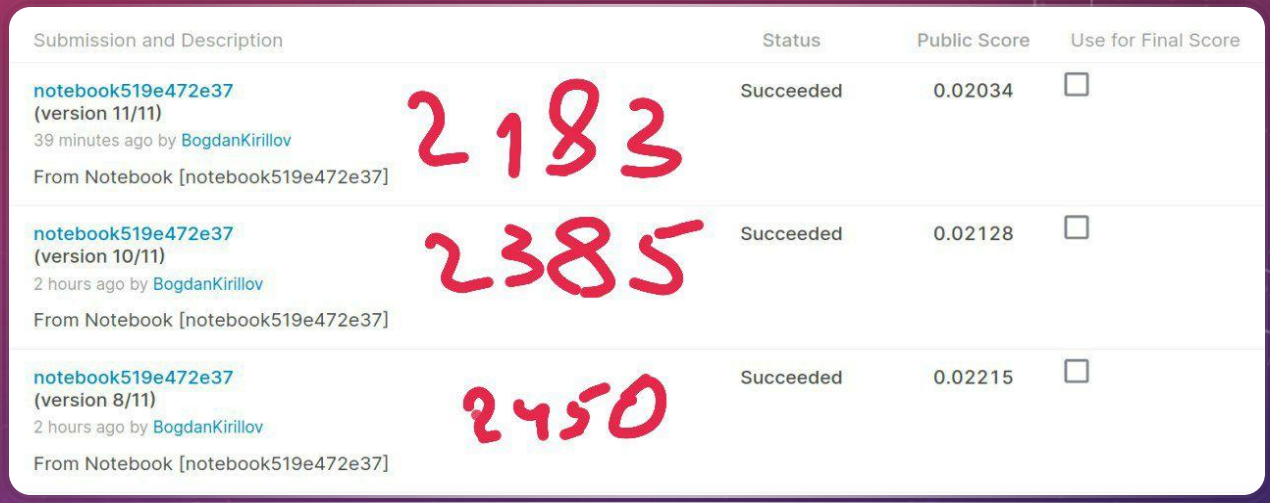
The code to reproduce the project is available at https://github.com/bakirillov/skoltech-mtf2020 and at https://www.kaggle.com/k1r1ll0v/notebook519e472e37?scriptVersionId=45291943

DIABETES

- Model configuration: 2 order Tensor layer (rank 5 Tucker decomposition) + dense layer + sigmoid
- Results: 0.75 - balanced accuracy, 0.7 - ROC-AUC

MECHANISMS OF ACTION PREDICTION

- Predicting the mechanism of action reveals the biological activity of a drug
- Task is to predict the probabilities for 206 possible mechanisms of action:
  - 'acetylcholinesterase_inhibitor'
  - 'ubiquitin_specific_protease_inhibitor'
  - vitamin_d_receptor_agonist'
  - ……..



The model used only one tensor layer, 3 order, Nx879x206, Tucker rank of 5 followed by sigmoid. No dense layers were used here.
The best attempt so far was trained for 60 epochs via Adam, with stepwise learning rate schedule starting from 0.001 and multiplied by 0.5 every fifth epoch.

CONCLUSION

We developed a model

We modified it with some constraints

We got some practiacl results

Bogdan Kirillov
Tg: @k0t3k_l4pk3
Bogdan.Kirillov@skoltech.ru

Anton Dmitriev
Tg: @addmitriev
Anton.Dmitriev@skoltech.ru