The following quizzes are aimed to check your knowledge of basic ML algorithms. Not all the information was covered in video lectures and we expect that you, as pointed in prerequisites, remember what is RF, GBDT, kNN.

If you don't know these abbreviations – you can check your Glossary (under Supplementary Materials tab).

You may find helpful these links while solving the quizz:

- Explanation of Random Forest
- Explanation/Demonstration of Gradient Boosting
- Example of kNN

Question 1

What back-propagation is usually used for in neural networks?

Correct answer:

- To calculate gradient of the loss function with respect to the parameters of the network

Incorrect answers:

- To propagate signal through network from input to output only. This is called "forward pass"
- Make several random perturbations of parameters and go back to the best one. This one doesn't involve gradients and have nothing to do with back-propagation
- Select gradient update direction by flipping a coin. In back-propagation gradients are calculated exactly, not random

Question 2

Suppose we've trained a RandomForest model with 100 trees. Consider two cases:

1. We drop the first tree in the model
2. We drop the last tree in the model

We then compare models performance on the train set. Select the right answer.

Correct answers:

• In the case1 performance will be roughly the same as in the case2. In RandomForest model we average 100 similar performing trees, trained independently. So the order of trees does not matter in RandomForest and performance drop will be very similar on average.

Incorrect answers:

• In the case1 performance will drop more than in the case2. In RandomForest model we average 100 similar performing trees, trained independently. So the order of trees does not matter in RandomForest.

• In the case1 performance will drop less than in the case2. Similar to the previous one.

Question 3

Suppose we've trained a GBDT model with 100 trees with a fairly high learning rate. Consider two cases:

1. We drop the first tree in the model
2. We drop the last tree in the model

We then compare models performance on the train set. Select the right answer.

Correct answers:

• In the case1 performance will drop more than in the case2. In GBDT model we have sequence of trees, each improve predictions of all previous. So, if we drop first tree – sum of all the rest trees will be biased and overall performance should drop. If

we drop the last tree -- sum of all previous tree won't be affected, so performance will change insignificantly (in case we have enough trees)

Incorrect answers:

- In the case1 performance will drop less than in the case2.
- In the case1 performance will be roughly the same as in the case2.

Question 4

Consider the two cases:

1. We fit two RandomForestClassifiers 500 trees each and average their predicted probabilities on the test set.
2. We fit a RandomForestClassifier with 1000 trees and use it to get test set probabilities.

All hyperparameters except number of trees are the same for all models.Select the right answer.

Correct answers:

- The quality of predictions in the case1 will be roughly the same as the quality of the predictions in the case2. Each tree in forest is independent from the others, so two RF with 500 trees is essentially the same as single RF model with 1000 trees

Incorrect answers:

- The quality of predictions in the case1 will be higher than the quality of the predictions in the case2.
- The quality of predictions in the case1 will be lower than the quality of the predictions in the case2.

Question 5

What model was most probably used to produce such decision surface? Color (from white to purple) shows predicted probability for a point to be of class "red".

Correct answers:

• Decision Tree. Decision surface consists of lines parallel to the axis and it is sharp.

Incorrect answers:

• Linear model. Decision surface is not linear.

• Random Forest. Decision surface consists of lines parallel to the axis and it is sharp -- in case of RF boundaries should be much more shooth.

• k-NN. Decision surface doesn't depend on distance from objects

Question 6

What model was most probably used to produce such decision surface?

Correct answers:

• Random Forest. Decision surface consists of lines parallel to the axis and its boundaries are smooth

Incorrect answers:

• Linear model. Decision surface is not linear

• Decision Tree. Decision surface consists of lines parallel to the axis and it is *not* sharp

• k-NN. Decision surface doesn't depend on distance from objects

Overview of methods

• Scikit-Learn (or sklearn) library

• Overview of k-NN (sklearn's documentation)

• Overview of Linear Models (sklearn's documentation)

• Overview of Decision Trees (sklearn's documentation)

• Overview of algorithms and parameters in H2O documentation

Additional Tools

• Vowpal Wabbit repository

- XGBoost repository
- LightGBM repository
- Interactive demo of simple feed-forward Neural Net
- Frameworks for Neural Nets: Keras,PyTorch,TensorFlow,MXNet, Lasagne
- Example from sklearn with different decision surfaces
- Arbitrary order factorization machines

AWS spot option:

- Overview of Spot mechanism
- Spot Setup Guide

Stack and packages:

- Basic SciPy stack (ipython, numpy, pandas, matplotlib)
- Jupyter Notebook
- Stand-alone python tSNE package
- Libraries to work with sparse CTR-like data: LibFM, LibFFM
- Another tree-based method: RGF (implemetation, paper)
- Python distribution with all-included packages: Anaconda
- Blog "datas-frame" (contains posts about effective Pandas usage)

Feature preprocessing

- Preprocessing in Sklearn
- Andrew NG about gradient descent and feature scaling
- Feature Scaling and the effect of standardization for machine learning algorithms

Feature generation

- Discover Feature Engineering, How to Engineer Features and How to Get Good at It
- Discussion of feature engineering on Quora

Question 1

Select true statements about n-grams.

Correct answers:

• N-grams can help utilize local context around each word. Correct, because ngrams encode sequences of words.

• N-grams features are typically sparse. Correct. Ngrams deal with counts of words occurrences, and not every word can be found in a document. For example, if we count occurrences of words from an english dictionary in our everyday speech, a lot of words won't be there, and that is sparsity.

Incorrect answers:

• N-grams always help increase significance of important words. No, ngrams deals with words occurrences and not their importance.

• Levenshteining should always be applied before computing n-grams. Although, there is Levenshtein distance, there is no such thing as Levenshteining.

Question 2

Select true statements.

Correct answers:

• Bag of words usually produces longer vectors than Word2vec. Correct! Number of features in Bag of words approach is usually equal to number of unique words, while number of features in w2v is restricted to a constant, like 300 or so.

• Semantically similar words usually have similar word2vec embeddings. Correct. This is one of the main benefits of w2v in competitions.

Incorrect answers:

• Meaning of each value in BOW matrix is unknown. Incorrect. Meaning of a value in BOW matrix is the number of a word's occurrences in a document.

• You do not need bag of words features in a competition if you have word2vec features. Incorrect. Both approaches are valuable and you should try to utilize both of them.

Question 3

Suppose in a new competition we are given a dataset of 2D medical images. We want to extract image descriptors from a hidden layer of a neural network pretrained on the ImageNet dataset. We will then use extracted descriptors to train a simple logistic regression model to classify images from our dataset.

We consider to use two networks: ResNet-50 with imagenet accuracy of X and VGG-16 with imageNet accuracy of Y (X < Y). Select true statements.

Correct answers:

• It is not clear what descriptors are better on our dataset. We should evaluate both. Correct! This depends on the a specific dataset and a specific task, so you should evaluate both!

Incorrect answers:

• With one pretrained CNN model you can get only one vector of descriptors for an image. Incorrect. With one CNN you can get different descriptors from different layers.

• Descriptors from ResNet 50 will always be better than the ones from VGG-16 in our pipeline. Incorrect. Although, ResNet50 shows better performance on Imagenet, this depends on the a specific dataset and a specific task.

• For any image descriptors from the last hidden layer of ResNet-50 are the same as the descriptors from the last hidden layer of VGG-16. Incorrect in general. Moreover it is hard to come up with an image that will have the same descriptors in both networks.

• Descriptors from ResNet-50 and from VGG-16 are always very similar in cosine distance. Incorrect. This depends on the a specific dataset and a specific task.

Question 4

Data augmentation can be used at (1) train time (2) test time

Correct answer:

True, True. Data augmentation can be used (1) to increase the amount of training data and (2) to average predictions for one augmented sample.

Feature extraction from text

Bag of words

- Feature extraction from text with Sklearn
- More examples of using Sklearn

Word2vec

- Tutorial to Word2vec
- Tutorial to word2vec usage
- Text Classification With Word2Vec
- Introduction to Word Embedding Models with Word2Vec

NLP Libraries

- NLTK
- TextBlob

Feature extraction from images

Pretrained models

- Using pretrained models in Keras
- Image classification with a pre-trained deep neural network

Finetuning

- How to Retrain Inception's Final Layer for New Categories in Tensorflow
- Fine-tuning Deep Learning Models in Keras