Thasina Tabashum

HW #8: Fashion MNIST tutorial breakdown (2 pts) Deep Learning

Due before taking Exam II

In this assignment we will step through the structure and design decisions of the Fashion MNIST tutorial given the context of the past month, also functioning as a review of the material we have covered. The goal is to understand the implications of every design choice throughout the structure and training of the network. Also, even if aware of the answer, it is important to be able to phrase your understanding appropriately.

**How to turn in the assignment:**

Given this is primarily for examination preparation, you are strongly encouraged to discuss all aspects of this assignment with others in the course, but all **submissions should be individually**.

The questions are numbered below - also use those numbers when providing the answers. Answers should be on a sheet and submitted as a **PDF document**. It is acceptable to only submit the answers without the associated questions for your convenience.

When you are finished, **submit through CANVAS** before the exam date.

Questions relating to the model in
https://www.tensorflow.org/tutorials/keras/classification

1. How many samples and how many total features for each sample are there in the entire data set from the tutorial? (note, this is the first question I often try to figure out when getting into details on ML consultations)

   Ans:

   Train+Test = (60k+10k)=70k samples and feature was (28*28=784 features)

2. How many total features would each sample have if the images were color instead of black and white?

   Ans: each sample feature: (28*28*3=2352)

3. How is the input data normalized?

   Ans: By dividing 255 for image input data.

4. The activation function (output nonlinearity) for the hidden layer units is ‘relu’: Describe the shape of that nonlinearity in words.

Ans: Nonlinear — When the activation function is non-linear, then a two-layer neural network can be proven to be a universal function approximator. The identity activation function does not satisfy this property. When multiple layers use the identity activation function, the entire network is equivalent to a single-layer model.

ReLU : A Rectified Linear Unit (A unit employing the rectifier is also called a rectified linear unit ReLU) has output 0 if the input is less than 0, and raw output otherwise. That is, if the input is greater than 0, the output is equal to the input. The operation of ReLU is closer to the way our biological neurons work.

5. The hidden layer is represented with 128 fully-connected neurons. Given the size of the images, how many parameters must be learned for this layer?

   Ans: 128*784 +128= 100480 parameters must be learned.

6. Similar question, the output layer is 10 fully-connected neurons, so how many parameters must be learned in this layer.

   Ans: 128*10+10=1290 paramaters.

7. If each parameter is 32 bits, how many megabytes do you need to store this model?

   Ans: (100480+1290)*32=3,256,640 bits/(1024*1024)=3.11mb)

8. Having 128 neurons in the hidden layer was likely found to perform best for this task. If we used dropout regularization for this layer, would it be better to consider increasing the number of neurons or decreasing them. Explain.

   Ans: If we use dropout layer it will certainly cancel some neurons. So,if 128 neurons perform best for this task, we need to increase neurons because we will lose neurons for dropout. We are already doing regularization so we can make the the model complex without having in mind that we are overfitting

9. Give your opinion if a convolutional neural network would be better or worse for this problem, and succinctly justify it.

   Ans: A convolutional neural network obviously perform best, because it will decrease the parameter size and eventually prevent overfitting, as a result it will perform better on test set not only in the training set.

10. Which of the lines (#1-5) below is closest to a description of this neural network.

| Output Type | Output Distribution | Output Layer | Cost Function |
|---|---|---|---|
| Binary | Bernoulli | Sigmoid | Binary cross-entropy |
| Discrete | Multinoulli | Softmax | Discrete cross-entropy |
| Continuous | Gaussian | Linear | Gaussian cross-entropy (MSE) |
| Continuous | Mixture of Gaussian | Mixture Density | Cross-entropy |
| Continuous | Arbitrary | See part III: GAN, VAE, FVBN | Various |

Answer:#2

11. The cost function here is a type of cross entropy. What other loss functions are possible that you've seen in the class?

    Ans: MAE( mean absolute error),MSE(mean square error)

12. The optimizer used here is Adam. Briefly describe what makes Adam different from standard gradient descent. No equations - just 1-2 sentences describing.

    Ans: Adam accumulates  adaptive learning rate(RMSPROP) and adaptive momentum. It has frequent update while Standard gradient descent is doing the update infrequently.

13. The metric being optimized here is accuracy. Name at least 2 other alternate metrics and succinctly describe them.

    Ans:

    1. True Positive (Recall)

    The True Positive Rate also called Recall is the go-to performance measure in binary/non-binary classification problems. Most if not all the time, we are only interested in correctly predicting one class. For example, if you were predicting diabetes, you will care more about predicting whether this person has diabetes than predicting this person does not have diabetes. In this situation, the positive class is 'this person has diabetes' and the negative class is 'this person does not have diabetes'. It is merely the accuracy of predicting the positive class (This is not the Accuracy performance metric. See number 4 below for more detail)

    2. ROC Curve (Receiver Operating Characteristic Curve)

    An ROC Curve shows the performance of your classification model at different thresholds (probability of classification into a certain class). It plots the True Positive Rate and False Positive Rate against each other. Lowering the threshold will increase your True Positive Rate but sacrifice your False Positive Rate and vice versa.

14. What is the batch size that Keras uses in the tutorial? (you'll have to look this up)

    Ans: 32 is the default batch size for keras.

15. If we increased the number of hidden layer units, what would happen to the training set accuracy and the test set accuracy compared to the original version?

    Ans: training set accuracy will higher and the test set accuracy will lower because there will be more parameter for hidden layer units and it will lead to overfitting.

16. Describe the one-hot encoding scheme in the last layer.

    Ans: It is quite common to use a One-Hot representation for categorical data in machine learning, for example textual instances in Natural Language Processing tasks. In Keras, the Embedding layer automatically takes inputs with the category indices (such as [5, 3, 1, 5]) and converts them into dense vectors of some length (e.g. 5 → [0.2 1.7 3.2 -7.6 ...]). What actually happens internally is that 5 gets converted to a one-hot vector (like [0 0 0 0 0 1 0 0 ... 0] of length equal to the vocabulary size), and is then multiplied by a normal weight matrix (such as a Dense layer), essentially picking the 5th indexed row from the weight matrix. However, there is no way in Keras to just get a one-hot vector as the output of a layer

17. Describe approximately what the softmax function does to the outputs of the 10 neurons in the output layer. In particular, what do the results of the softmax function on those outputs represent?

    Ans: Softmax extends this idea into a multi-class world. That is, Softmax assigns decimal probabilities to each class in a multi-class problem. Those decimal probabilities must add up to 1.0. This additional constraint helps training converge more quickly than it otherwise would.

18. What is the nonlinearity used in the output neurons prior to the application of softmax

    Ans: Sigmoid function is used

19. Why might it be better to not have ‘relu’ as the activation function for the output layer?

    Ans:ReLU : A Rectified Linear Unit (A unit employing the rectifier is also called a rectified linear unit ReLU) has output 0 if the input is less than 0, and raw output otherwise. That is, if the input is greater than 0, the output is equal to the input. The operation of ReLU is closer to the way our biological neurons work. So, if we use relu we will not get output like 0 to 1 like softmax which create more sense for classification.

20. Is this tutorial using cross-validation?Ans: no