Practice for Final Exam - Applied Machine Learning COMS W4995

Date:

Name:

UNI:

For all choice boxes, please fill in the box you want to choose like this: ■
Otherwise your answer can not be graded.

1 True/False (+2pt each)

False

	True	False
Given a trained word2vec CBOW model, it's easy to compute the vectors for out-of-vocabulary word.	☐	■
In Latent Dirichlet Allocation, each document is assigned a single topic.	☐	■
You can always extract as many principal components as there are input features.	☐	■
Adding a batch normalization layer increases the number of parameters in a neural network.	■	☐
A partial dependence plot of a linear model will always be linear.	■	☐
A Gaussian Mixture Model allows evaluating the probability of a new point under a fitted model.	■	☐
Randomized search is less effective than grid search in finding good settings in high-dimensional parameter spaces.	☐	■
Isolation Forests assume Gaussian Distributed Data	☐	■
In a bag-of-word model with unigrams, using stop-words will reduce the number of features only marginally.	■	☐
Convolutional layers in a neural network typically have less parameters than densely connected layers.	■	☐

2 Multiple choice (20pt)

Select all choices that apply.

2.1 2.1 Which of the following statements apply to neural networks?

- ☐ Fast to train on large datasets.
- ■ Can learn arbitrarily complex functions.
- ☐ Work well when little training data is available.
- ■ Provide state-of-the-art performance in computer vision and audio analysis.
- ☐ Have no hyper-parameters to tune.

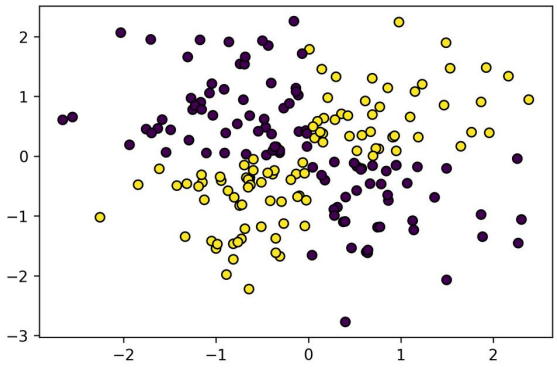
2.2 Which of the following models requires solving an optimization problem (as opposed to a closed-form formula) to transform data after the model is fitted?

- ■ Non-Negative Matrix Factorization
- ■ Latent Dirichlet Allocation
- ☐ PCA
- ☐ Linear Discriminant Analysis
- ■ Paragraph Vectors

2.3 Given a dataset including multiple copies of the most informative feature. Which of the following methods will guarantee to identify at least one of the copies as highly informative? Assume the relationship is linear.

- ■ Univariate statistics
- ☐ Permutation importance of a random forest
- ■ Lasso coefficients
- ☐ Gini importance of a decision tree
- ■ Sequential feature selection with gradient boosted trees
- ☐ Ridge regression coefficients

2.4 Given a two-class classification dataset with the two features shown below and additional non-informative features, which of the following feature selection methods would be able to identify the these two features as informative?



- ☐ SelectPercentile(f_classif)
- ■ SelectKBest(mutual_info_classif)
- ■ SelectFromModel(DecisionTreeClassifier())
- ■ SequentialFeatureSelector(SVC(kernel='rbf'))
- ☐ RFE(LogisticRegression())

3 Debugging (10pt each)

For each code snippet, find and explain all errors given the task. Assume all necessary imports have been made. There can be more than one error per task!

3.1 Task: Perform grid-search on a Keras Sequential model for the number of units (50, 100 or 200) in the hidden layer. The network should be a one-hidden-layer network for 64 input features and 8 classes.

```
 1 | X_train, X_test, y_train, y_test = train_test_split(X, y)
 2 | model = Sequential([Dense(50),
 3 |                                    Dense(8,
activation=”softmax”))
 4 |
 5 | model.compile(“adam”, “multiclass_crossentropy”,
metrics=[“accuracy”])
 6 |
 7 | param_grid = {‘hidden_units’: [50, 100, 200]}
 8 | grid = GridSearchCV(model, param_grid)
 9 | grid.fit(X_train, y_train)
10 | score = grid.score(X_test, y_test)
```

Line 2: No input shape defined.
Line 2: No non-linear activation function specified.
Line 8: Can’t use “sequential” in a GridSearchCV, need to define a callable and give it to KerasClassifier.

3.2 Task: Write down the computation in a forward-pass of a feed-forward neural network for classification with one hidden layer with 100 units, tanh non-linearity and a drop-out rate of 50% on the hidden layer.

```
1 |def forward(X, w1, b1, w2, b2):
2 |    h1_net = np.dot(X, w1 + b1)
3 |    dropout_mask = np.random.uniform(size=100) > .5
4 |    h1_net[dropout_mask] = 0
5 |    h1 = np.tanh(h1_net)
6 |    out_net = np.dot(X, w2 + b2)
7 |    out_exp = np.exp(out_net)
8 |    return out_exp - np.sum(out_exp)
```

Line 2, 6: Bias should be added after matrix multiplication.
Line 8: Need to divide by np.sum(out_exp) for softmax.
Line 6: Out_net should use h1 not X.

4 Coding (10 each)

Assume all necessary imports have been made.

4.1 Define a multi-layer perceptron using the Keras Sequential interface with relu non-linearity and a single hidden layer with 100 hidden units for classifying the iris dataset.

```
Sequential([Dense(100, input_shape=(4,), activation="relu"),
            Dense(3, activation="softmax")])
```

4.2 Apply PCA to detect outliers in a dataset given as X by reducing it to 10 dimensions. Assume there are 5% outliers. Include preprocessing. Assume all necessary imports are made.

```
X_scaled = scale(X)
pca = PCA(n_components=10).fit(X)
X_reconstruction = pca.inverse_transform(pca.transform(X))
error = ((X_scaled - X_reconstruction) ** 2).sum(axis=1)
threshold = np.percentile(error, 5)
outlier = error > threshold
```

5 Concepts (5pt each)

Answer each question with a short (2-5 sentences) explanation.

5.1 Explain the “CBOW” approach used in word2vec. How are the word representations found?
Given the context of a word in one-hot encoding, try to predict the missing word. Prediction is done using two matrix multiplications (like a linear neural net), where the hidden layer corresponds to the size of the embedding vectors. The prediction is done using softmax. The model is learned using SGD sampling words and contexts from the training data.

5.2 Explain how “batch normalization” works.
During the forward pass in a neural network, before the non-linearity, the activations are normalized to have zero mean and unit variance. This is done separately for each mini-batch, and re-computed for each iteration. An additional scaling factor and offset is learned on the standardized activations to maintain the same representational power. The gradient computation takes the normalization into account. Normalizing activations in this way leads to faster learning.

5.3 Compute the number of parameters in a convolutional neural network with 16x16x1 input, followed by two 3x3 convolution layers with 4 maps each, followed by a 2x2 max pooling layer followed by an output layer with two units (don’t forget biases). You can just write out the multiplications and additions for each layer, you don’t need to compute the additions and multiplications.
1 * 3 * 3 * 4 + 4 filter + bias for first conv layer
+ 4 * 3 *3 * 4 + 4 filter + bias for second conv layer
+ 6 * 6 * 4 * 2 + 2 dense layer weights + biases. Resolution after first layer is 14, second 12, pooling 6 -> 6 * 6 * 4 hidden units.
Note: The question was ambiguous, this solution assumed “valid” convolutions. If you assume “same” convolutions the number of weights in the last layer is 8 * 8 * 4 * 2.

5.4 Explain what successive halving is used for in machine learning and how it works.
Successive halving is a strategy for hyper parameter and model selection that can be more efficient than grid-search and randomized search. Successive halving trains models on increasing subsets of the dataset, retaining only well performing parameter configurations.
Starting from some budget r for the first iteration (say 1000 samples) and an initial pool of configurations, in each iteration the procedure keeps the best half of the configurations according to a chosen criterion (say accuracy), and doubles the budget for the next iteration, i.e. doubles the number of samples used for training. Instead of using doubling and halving, using 3 or 4 as basis is also possible.

6. Bonus question (there won't be one in the exam)!
What TV shows have been referenced in the slides and homeworks of this course?

- ☐ Firefly
- ■ Archer
- ■ Rick and Morty
- ☐ Steven Universe
- ■ Hitchhiker's guide to the galaxy
- ☐ One Punch Man