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.	■	☐
ARI is a practical way of adjusting the number of clusters in K-Means for exploratory data analysis.	☐	■
A Gaussian Mixture Model allows evaluating the probability of a new point under a fitted model.	■	☐
The normalized mutual information (NMI) is not defined for cluster assignments with different numbers of clusters.	☐	■
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?
- ■ Non-Negative Matrix Factorization
- ■ Latent Dirichlet Allocation
- □ PCA
- □ Linear Discriminant Analysis
- ■ Paragraph Vectors

2.3 What are reasons to prefer Non-negative Matrix Factorization over PCA?
- □ Better reconstruction of the data.
- ■ Sign of the components is meaningful.
- ■ No cancellation effects.
- ■ Can extract non-linear features.
- □ Faster.
- □ Deterministic results.

2.4 Which of the following cluster evaluation methods are unsupervised?

- ■ Silhouette Score
- □ Adjusted Rand Index
- □ Normalized Mutual Information
- ■ Stability based score

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.

```
X_train, X_test, y_train, y_test = train_test_split(X, y)
model = Sequential([Dense(50),
                    Dense(8, activation="softmax"))

model.compile("adam", "multiclass_crossentropy",
metrics=["accuracy"])

param_grid = {'hidden_units': [50, 100, 200]}
grid = GridSearchCV(model, param_grid)
grid.fit(X_train, y_train)
score = grid.score(X_test, y_test)
```

No input shape defined.
No non-linear activation function specified.
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.

```python
def forward(X, w1, b1, w2, b2):
    h1_net = np.dot(X, w1 + b1)
    dropout_mask = np.random.uniform(100) > .5
    h1_net[dropout_mask] = 0
    h1 = np.tanh(h1_net)
    out_net = np.dot(X, w2 + b2)
    out_exp = np.exp(out_net)
    return out_exp - np.sum(out_exp)
```

Bias should be added after matrix multiplication.
Need to divide by np.sum(out_exp) for softmax.

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.
# 1 for getting first two layer sizes right, 1 for filter sizes 1 for bias 2 for last layer size

5.4 Explain the generative process for a document in Latent Dirichlet Allocation.
For each of k topics, a topic-word distribution is drawn from a dirichlet distribution. For each document in the corpus, a distribution over topics is drawn from a dirichlet distribution. For each word in a document, a topic is chosen according to the document specific topic distribution. Then the word is drawn from the topic-word distribution for this topic.

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
- ☐ Altered Carbon
- ■ One Punch Man