COMP0090 2020/21 Assignment 3: “Convoluted bag”

Pontus SAITO STENETORP
p.stenetorp@cs.ucl.ac.uk

1 Instructions

For this assignment the maximum number of points obtainable is 100, which maps one-to-one for the mark given for the assignment. The assignment are to be carried out in groups of *up to five students*, which may or may not be the same groups as those for future assignments. You are free to use *any programming language and libraries/frameworks* of your choice – also to freely change between them for different tasks.

The assignment is *due by Friday, December 18th 2020 at 16:00 (Europe/London)*[1] and is to be submitted via Moodle. You shall use LaTeX[2] to produce *a single PDF document*. Use the following template as a starting point: `https://www.overleaf.com/read/gyphwtvffnxj`, but you are free to improve upon it as you see fit.

Describe briefly (about a paragraph) in your submission *which group member contributed to which part to the assignment*. It is expected that each group member contributes towards at least one of the tasks, but as with all group work *all members are collectively responsible for the submission in its entirety* and should confirm that they are familiar with all solutions prior to submission.

Kindly report errors (even typos) and ask for clarifications when needed, this assignment is to be an exercise in deep learning, not mind reading. Corrections to this assignment post-release will be listed in Section 4.

[1] The module organisers are awaiting approval for an extension.

[2] `https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes`

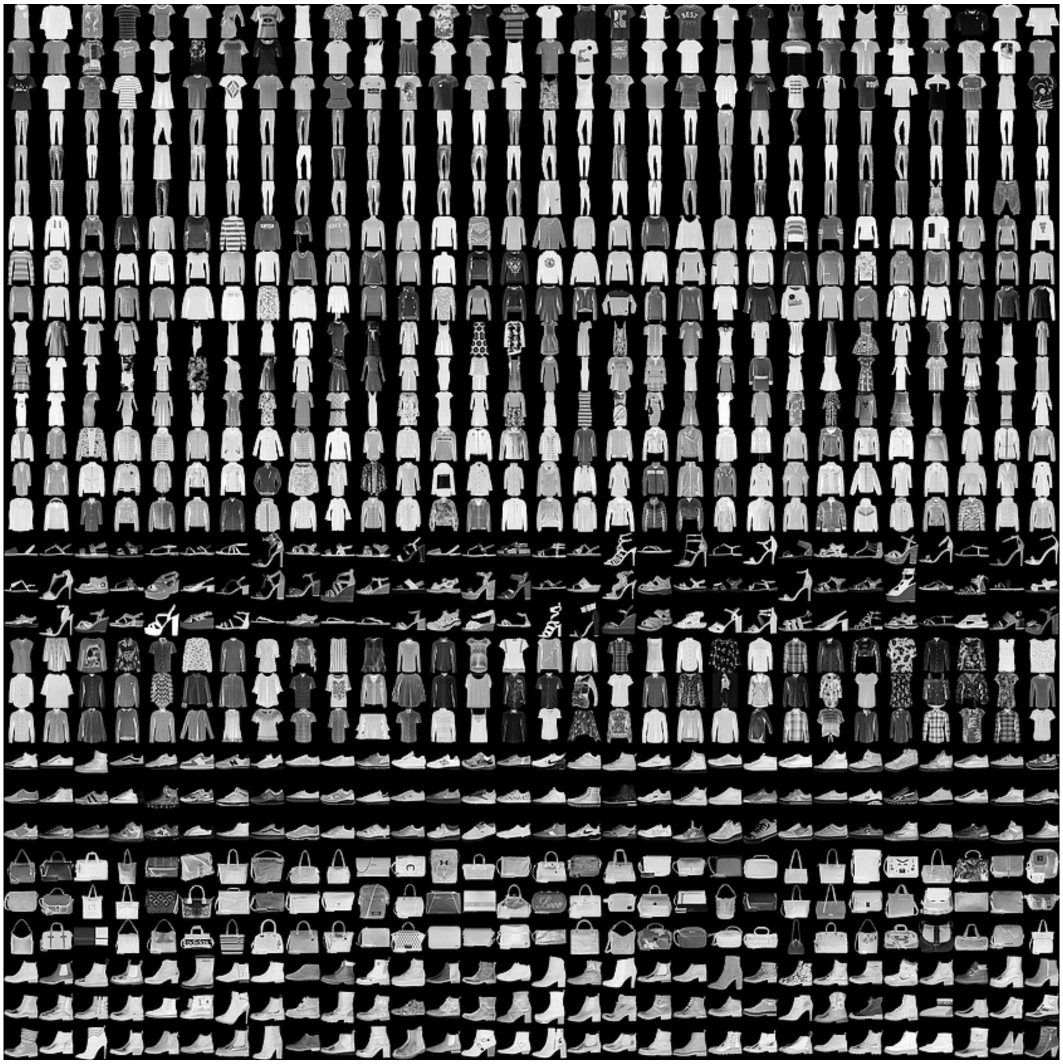

Figure 1: Examples from each of ten classes from Xiao et al. (2017), each class occupies three rows.

2 Data

Throughout this assignment we will use a dataset which at this point you should be very familiar with, *we will however make some minor alterations to it*.

A standard dataset in machine learning is the MNIST dataset of handwritten digits that has for many years been used to train and evaluate machine learning algorithms. However, even simple algorithms can perform relatively well on MNIST and its usage has increasingly come into question. Zalando[3] is a German e-commerce company and that has released a dataset "Fashion-MNIST" (Xiao et al., 2017) which instead of digits from 0 to 9, contains 28-by-28 pixel, greyscale images of ten distinct fashion items (see Figure 1 for an example subset).

As two tasks for this assignment will involve pre-training and transfer learning, we need to make some minor modifications to our "vanilla" dataset to create two subsets that do not share labels. Thus we split Fashion-MNIST into Fashion-MNIST-1 and Fashion-MNIST-2, which are of equal size and both contains a disjoint half of the original labels each.

Code for loading the data can be found in the Colaboratory notebook for the assignment: `https://colab.research.google.com/drive/1T5a6LrNA03tUxGgEfJvnUVz1vg3TX9g3`.

[3] `https://www.zalando.de`

3 Tasks

3.1 Convolutional neural network *(20 points)*

We are finally ready to employ a computer vision model for our long-time computer vision dataset – a perfect fit. For this task you may want to adapt the convolutional neural network code from the lecture notebook.

To complete this task you are expected to:

1. Implement a multi-class, convolutional neural network with cross-entropy loss for the Fashion-MNIST-1 data.
2. Train your final model to convergence on the training set using an optimisation algorithm of your choice.
3. Provide a plot of the loss on the training set and validation set for each epoch of training.
4. Provide the final accuracy on the training, validation, and test set.
5. Analyse the errors of your models by constructing a confusion matrix.[4] Which classes are easily "confused" by the model? Hypothesise why.

[4] `https://en.wikipedia.org/wiki/Confusion_matrix`

		Layers					Loss			Accuracy		
	Name	Convolution	Classification	Optimiser	L. rate	Regularisation	Train	Valid.	Test	Train	Valid.	Test
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												

Table 1: Example table to complete for the task in Section 3.2.

3.2 Convolutional neural network variants *(20 points)*

A standard procedure in machine learning research to understand both your data and model is to explore variants that you hypothesise might work better based on experimental observations.

To complete this task you are expected to:

1. Implement a multi-class, convolutional neural network with cross-entropy loss for the Fashion-MNIST-1 data, then fill in information regarding it into a table akin to Table 1.[5]
2. Iteratively make modifications to your model based on how your changes affect the *validation loss*, try to minimise it by producing nine additional variants.[6] Note that you will need to construct one loss function without regularisation and one with regularisation, the former to obtain the loss to enter into your table and the latter to obtain your gradients, or your losses will not be comparable as you change the regulariser. It is also a good idea to plot the training and validation loss across epochs, rather than simply observing the final validation loss as the shape of the curves provide further insights into the model performance.
3. Was the lowest *test loss* obtained for model with the lowest *validation loss*? If not, why do you think this was the case?

[5] You are allowed to consider alternative attributes to those in Table 1, it is simply an example.

[6] You are free to construct more variants, but the task is considered completed after you have obtained the results for ten variants.

3.3 Feature map inspection *(20 points)*

It is well established that convolutional neural networks learn interpretable feature maps and for this task we will explore the feature maps learnt for a given model on our dataset.

To complete this task you are expected to:

1. Implement a multi-class, convolutional neural network with cross-entropy loss for the Fashion-MNIST-1 data.
2. Train your final model to convergence on the training set using an optimisation algorithm of your choice.
3. Retrieve and visualise the feature maps for each layer of your convolutional neural network.
4. Qualitatively analyse the feature maps and hypothesise what they capture and if possible – in particular for the deeper layers – associate them with the output classes.

3.4 Pre-training *(20 points)*

We have previously explored autoencoders as generative models, however, they can also be used to initialise the weights for a predictive model that share the same structure as the *encoder* of the autoencoder. This can be especially useful when the amount of training data is limited, but you have access to a large amount of unlabeled data which shares structure with your labeled data.

To complete this task you are expected to:

1. Implement an autoencoder with mean squared error loss for the *Fashion-MNIST-1* and *Fashion-MNIST-2* data.
2. Train your model to convergence on the combined *training*, *validation*, and *test* set of *Fashion-MNIST-2* and *training* set of *Fashion-MNIST-1* using an optimisation algorithm of your choice.
3. Implement a multi-class, multi-layer perceptron with cross-entropy loss for the *Fashion-MNIST-1* data, which shares the same structure as the encoder of your autoencoder.
4. Compare using *random weights* to those from your autoencoder to initialise the multi-layer perceptron by plotting the training and validation loss for both options when you use 5%, 10%, ..., 100% of the available *Fashion-MNIST-1* training data to train your model. Is there a point where one initialisation option is superior to the other? Is one option always superior to the other?
5. Provide the final accuracy on the training, validation, and test set for the best model you obtained for each of the two initialisation strategies.

Optionally you may implement a convolutional autoencoder as opposed to a "vanilla" autoencoder for this task, which will allow you to use a convolutional neural network as opposed to a multi-class, multi-layer perceptron as your predictive model. This will yield no additional points, but some students may enjoy the implementation challenge and the fact that you then can use the same family of models throughout the entire assignment.

3.5 Transfer learning *(20 points)*

A core tenet in feature learning is that good learnt features shall be transferable between tasks. This has become a reality in computer vision where it is commonplace to use existing weights trained by others on a large-scale datasets such as ImageNet (Deng et al., 2009), remove the classification layers, and then train new classification layers using your own labelled data while preserving the same convolutional neural network structure. This is very much akin to pre-training as we saw in Section 3.4, but requires labels for both datasets. For this task we will emulate a similar setting with our Fashion-MNIST-1 and Fashion-MNIST-2 datasets.

To complete this task you are expected to:

1. Implement a multi-class, convolutional neural network with cross-entropy loss for the *Fashion-MNIST-2* data.
2. Iteratively tune your model structure and hyperparameters using the *validation set* of *Fashion-MNIST-2*, until you arrive at a model performance you are comfortable with.[7]
3. Implement a multi-class, convolutional neural network with cross-entropy loss for the *Fashion-MNIST-1* data, which shares the same structure as the one you used for the *Fashion-MNIST-2* data.
4. Compare using *random weights* to those obtained by training on *Fashion-MNIST-2* – you should randomly re-initialise the classification layer though – to initialise the multi-class, convolutional neural network by plotting the training and validation loss for both options when you use 5%, 10%, . . . , 100% of the available *Fashion-MNIST-1* training data to train your model. Is there a point where one initialisation option is superior to the other? Is one option always superior to the other?
5. Provide the final accuracy on the training, validation, and test set for the best model you obtained for each of the two initialisation strategies.

[7] For example, when you have improved upon your initial "guess" and the performance is somewhere close to what you have seen for the models for other tasks on *Fashion-MNIST-1* during this assignment.

4 Errata

Empty, for now…

References

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms, 2017.