DATA.ML.200 Pattern Recognition and Machine Learning

Exercise Set 5: November 23–November 27, 2020

- Exercises consist of both `pen&paper` and `python` assignments.
- Prepare a single PDF and return to Moodle on Friday, November 27th at 23:55 at the latest.
- Mark on 1st page which exercises you did.

1. `pen&paper` *Count the number of parameters in a **dense** neural network.*

   Consider the traditional shallow neural network architecture of Figure 1. Suppose our inputs are 128×128 RGB bitmaps of 10 categories.

   Let the network structure be the following:

   - The input size is $128 \times 128 \times 3$ and vectorized (flattened) before feeding to the net.
   - On the 1st layer there are 100 nodes (marked in blue)
   - On the 2nd layer there are 100 nodes (marked in blue)
   - On the 3rd (output) layer there are 10 nodes (marked in blue; one for each class)

   Compute the number of parameters (coefficients) in the net.

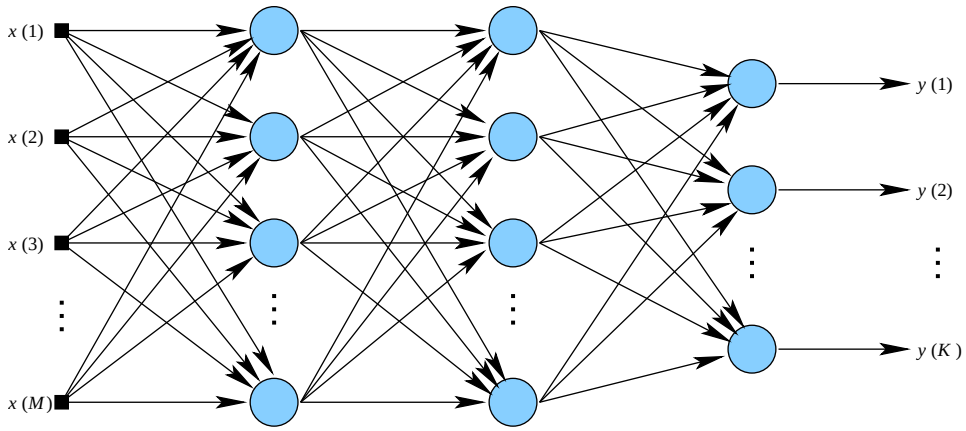

Figure 1: Vanilla neural network.

```
>>> model.summary()
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_12 (Conv2D)           (None, 128, 128, 32)      XXXXX
_________________________________________________________________
max_pooling2d_9 (MaxPooling2 (None, 64, 64, 32)        XXXXX
_________________________________________________________________
conv2d_13 (Conv2D)           (None, 64, 64, 32)        XXXXX
_________________________________________________________________
max_pooling2d_10 (MaxPooling (None, 32, 32, 32)        XXXXX
_________________________________________________________________
flatten_3 (Flatten)          (None, 32768)             XXXXX
_________________________________________________________________
dense_3 (Dense)              (None, 10)                XXXXX
=================================================================
Total params: XXXXX
Trainable params: XXXXX
Non-trainable params: 0
_________________________________________________________________
```

Figure 2: A CNN model summary as reported by Keras

2. **pen&paper** *Count the number of parameters in a **convolutional** neural network.*
   Consider the Keras model defined in Listing 2. All convolutional kernels are of size $w \times h = 5 \times 5$ and convolutions use "same" mode for padding.
   a) Draw a diagram of the network.
   b) Compute the number of parameters for each layer.
   c) How many scalar multiplications per image take place on the first convolutional layer?

3. **python** *Audio classification.*

   This week exercises consider auditory scene classification. The data is originally from this course competition in 2018[1]. Load competition data from one of the following sources:

```
Sharepoint: https://tinyurl.com/y2gp2jl3
Kaggle:     https://www.kaggle.com/c/acoustic-scene-2018/data
Dropbox:    https://www.dropbox.com/s/zd694bdomrqen3e/Acoustic.zip
```

   The data is readily preprocessed and split to training and testing for you. Inputs consist of spectrograms of shape `(40, 501)` (frequency, time), and there are total of 15 classes. See competition website for more details.

   Train a **Random forest classifier** for the data. As the data consist of matrices (spectrograms), you will need to vectorize the first.

   a) What is the accuracy with 10 trees?

   b) What is the accuracy with 50 trees?

   c) What is the accuracy with 100 trees?

   Note: The RF accuracy starts to saturate at about 500–1000 trees, so this is the amount you's normally use.

4. Train a convnet for the acoustic data. Treat the spectrograms as images and adjust the convnet of question 2 to this data:

   a) Add a dummy "color" dimension as TF expects to get 4-dimensional inputs: `X_train = X_train[..., np.newaxis]`.

   b) Convert `y_train` to one-hot-encoded.

   Run for at least 5 epochs (or more if you have a GPU) and report the accuracy.

5. Train a LSTM recurrent net for the acoustic data. The net should consist of:

   - One LSTM layer with 32 nodes.
   - One dense layer with softmax activation and 15 outputs.

   Run for at least 5 epochs (or more if you have a GPU) and report the accuracy.

[1] `https://www.kaggle.com/c/acoustic-scene-2018`