Engineering Tripos Part IIB FOURTH YEAR

Module 4F12: Computer Vision

Examples Paper 4

Straightforward questions are marked †
Tripos standard (but not necessarily Tripos length) questions are marked ∗

1. ∗ *Neural networks*

   (a) Consider a single neuron that takes a multi-dimentional input $\mathbf{z}$ forms the scalar product with a set of weights $\mathbf{w}$ and passes this through a sigmoid activation function,

   $$x(\mathbf{z};\mathbf{w}) = \frac{1}{1+\exp(-\mathbf{w}^{\mathrm{T}}\mathbf{z})}.$$

   The neuron can be trained on a set of example inputs $\{\mathbf{z^{(n)}}\}_{\mathbf{n=1}}^{\mathbf{N}}$ and outputs $\{\mathrm{t}^{(\mathrm{n})}\}_{\mathrm{n}=1}^{\mathrm{N}}$ by minimising the relative-entropy,

   $$G(\mathbf{w}) = -\sum_{\mathbf{n}} \left[\mathrm{t}^{(\mathrm{n})} \log \mathrm{x}(\mathbf{z^{(n)}};\mathbf{w}) + (1-\mathrm{t}^{(\mathrm{n})}) \log\left(1-\mathrm{x}(\mathbf{z^{(n)}};\mathbf{w})\right)\right].$$

   Show that the derivatives of the relative-entropy are given by,

   $$\frac{\mathrm{d}}{\mathrm{d}\mathbf{w}} G(\mathbf{w}) = -\sum_{\mathbf{n}} (\mathrm{t}^{(\mathrm{n})} - \mathrm{x}^{(\mathrm{n})})\mathbf{z^{(n)}}.$$

   The single neuron classifier is also called logistic regression.

   (b) Consider applying a neural network to scene classification in which images are assigned to one of a number of possible categories. The targets are now represented by a vector $\mathbf{t}$ which has a single element set to 1, indicating the correct scene, and all other values are set to 0. Consider a soft-max network in which the output of the network is a vector, $\mathbf{x}$, with elements given by a soft-max function,

   $$x_i(\mathbf{z};\mathbf{w}) = \frac{\exp(\mathbf{w_i}^{\mathrm{T}}\mathbf{z})}{\sum_{\mathbf{j}} \exp(\mathbf{w_j}^{\mathrm{T}}\mathbf{z})}.$$

      i. Interpretting the output of the network as $x_i = p(\mathrm{t_i}=1|\mathbf{w},\mathbf{z})$ write down a cost-function for training this network based on the log-probability of the training data given the weights $\mathbf{w}$ and inputs $\{\mathbf{z^{(n)}}\}_{\mathbf{n=1}}^{\mathbf{N}}$.

ii. What is the relationship between this network and the one described in the first part of this question?

2. $*$ *Convolutional neural networks for classification*

Consider applying a simple convolutional neural network to a 1D image. Let $z_i^{(n)}$ denote the nth image in the training set. The network has a single convolutional stage containing a single convolutional weight, w_k,

$$a_i^{(n)} = \sum_k w_k z_{i-k}^{(n)}.$$

The non-linear stage uses a point-wise non-linearity, $y_i = f(a_i)$. There is no pooling stage. The readout occurs via a logistic function that pools across the convolutional layer,

$$x^{(n)} = \frac{1}{1 + \exp(-\sum_i v_i y_i^{(n)})}.$$

The network can be trained on a set of example inputs $\{\mathbf{z^{(n)}}\}_{\mathbf{n=1}}^{\mathbf{N}}$ and outputs $\{\mathrm{t^{(n)}}\}_{\mathrm{n=1}}^{\mathrm{N}}$ by minimising the relative-entropy,

$$G(\mathbf{w}, \mathbf{v}) = -\sum_{\mathbf{n}} \left[\mathrm{t^{(n)}} \log \mathrm{x}(\mathbf{z^{(n)}}; \mathbf{w}, \mathbf{v}) + (1 - \mathrm{t^{(n)}}) \log \left(1 - \mathrm{x}(\mathbf{z^{(n)}}; \mathbf{w}, \mathbf{v}) \right) \right].$$

Show that the derivatives of the objective function with respect to the convolutional weights, w_k, can themselves be computed efficiently using convolutions.

3. $*$ *Convolutional neural networks for regression* (Tripos question 2015/16)

A convolutional neural network is to be used for estimating the age of a person from an image of their face. A labelled dataset has been collected that contains N greyscale face images $\{Z^{(n)}\}_{n=1}^{N}$ and labels $\{\mathrm{t^{(n)}}\}_{\mathrm{n=1}}^{\mathrm{N}}$ that are the age of the person in each image.

The network contains three stages. The first stage carries out a 2-D convolution between the image pixels $Z_{i,j}^{(n)}$ and convolutional weights $W_{i,j}$,

$$a_{i,j}^{(n)} = \sum_{k,l} W_{k,l} Z_{i-k,j-l}^{(n)}.$$

The second stage applies a point-wise non-linearity $y_{i,j}^{(n)} = f\left(a_{i,j}^{(n)}\right)$.

The third stage applies a set of output weights $V_{i,j}$ in order to form the scalar output of the network,

$$x^{(n)} = \sum_{i,j} V_{i,j} y_{i,j}^{(n)}.$$

The network's weights will be trained using the following objective function,

$$G(V,W) = \frac{1}{2\sigma^2}\sum_{n=1}^{N}\left(\mathrm{t}^{(\mathrm{n})} - \mathrm{x}^{(\mathrm{n})}\right)^2 + \frac{\alpha}{2}\sum_{i,j}V_{i,j}^2 + \frac{\beta}{2}\sum_{i,j}W_{i,j}^2.$$

(a) Provide an interpretation for the network's output in terms of probability distributions and use this to justify the form of the objective function.

(b) Describe how to train the network's convolutional weights W using gradient descent. Compute the derivative required to implement gradient descent. Simplify your expression and interpret the terms.

(c) Describe enhancements to the architecture of the network that might improve its ability to estimate the age of a person from an image of their face.

Richard E. Turner

4. ** CNN architectures for computer vision*

   You are given a dataset of 60000 images (32×32 pixels each) of digits and corresponding class labels (0, 1, ..., 9) and asked to build a CNN which performs digit classification.

   (a) Propose a simple architecture for such a network and explain the motivation behind your choice. Write down mathematical definitions of all layers used and explain their corresponding roles. Provide a calculation of network parameters in your proposed architecture.

   (b) Describe how would you train such a network. Include details of: (i) data preparation, (ii) objective function, (iii) optimization algorithm, (iv) weight initialization and (iv) key steps of performance tuning.

   (c) Which state-of-the-art CNN architecture would you use if you were required to train a network to classify images (224×224 pixels each) into one of 1000 classes using ImageNet dataset (contains pairs of images and corresponding class labels)? Explain key design properties of this architecture.

   (d) How would you amend the architecture of the network mentioned in section (c) and your training procedure proposed in section (b) if you were asked to build a network which performs semantic segmentation of objects of some 20 classes (e.g. dog, person, etc). You can assume that along with ImageNet dataset you also have Microsoft Common Objects in Context Dataset (containing pairs of images and corresponding per-pixel class labels) provided.

Ignas Budvytis