Engineering Tripos Part IIB FOURTH YEAR

Module 4F10: DEEP LEARNING AND STRUCTURED DATA

Examples Paper 2

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

Deep Learning

1. Residual networks are often used with very deep networks. For a single layer, layer l with N_l nodes, of a residual network the activation function can be expressed as (note no bias is used here)

$$\boldsymbol{y}^{(l)} = \boldsymbol{x}^{(l)} + \boldsymbol{\phi}\left(\mathbf{W}\boldsymbol{x}^{(l)}\right); \quad \boldsymbol{\phi}(\boldsymbol{z}) = \begin{bmatrix} 1/(1+\exp(-z_1)) \\ \vdots \\ 1/(1+\exp(-z_{N_l})) \end{bmatrix}$$

where $\boldsymbol{x}^{(l)}$ and $\boldsymbol{y}^{(l)}$ are the input and output to layer l, $\boldsymbol{\phi}()$ is a sigmoid activation function that operates on each element of the vector. Thus the standard activation function, $\boldsymbol{\phi}()$, has a residual connection added.

   (a) For this network configuration derive an expression for the derivative $\frac{\partial \boldsymbol{y}^{(l)}}{\partial \boldsymbol{x}^{(l)}}$.

   (b) Based on the answer in (a), comment on how the use of a residual connection can help in reducing the vanishing gradient problem.

   (c) Give the relationship between the input and output for this layer when the residual connection is replaced by a highway connection. Do you expect this form of connection to also help address the vanishing gradient problem?

2. A simplified recurrent neural network is to be trained. For this simplified network a linear activation function is used such that the recurrence relationship has the form

$$\boldsymbol{h}_t = \mathbf{W}\boldsymbol{h}_{t-1}$$

Show that the activation function output at time t can be expressed as

$$\boldsymbol{h}_t \approx \lambda^t \boldsymbol{q}\boldsymbol{v}^{\top}\boldsymbol{h}_0$$

when t gets large. Clearly define scalar λ, and vectors $\boldsymbol{q}$ and $\boldsymbol{v}$. What is the implication of this expression for training recurrent neural networks with long sequences?

3. Figure 3 shows a Gated Recurrent Unit (GRU) that is to be used for a K class classification problem. The output of the GRU at time t, $\boldsymbol{h}_t$, is used to predict a probability mass function (PMF) over the K classes at time t, $\boldsymbol{y}_t$, as well as the input to the next time instance.

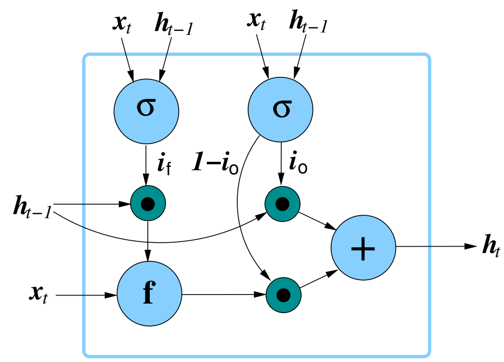

Figure 1: Gated Recurrent Unit

(a) What is an appropriate form for the gating activation function $\boldsymbol{\sigma}(\boldsymbol{x}_t, \boldsymbol{h}_{t-1})$? You should justify your answer.

(b) Give the overall expression for how the inputs, $\boldsymbol{x}_t$ and $\boldsymbol{h}_{t-1}$, are transformed to yield the output $\boldsymbol{y}_t$ and new history vector $\boldsymbol{h}_t$. You should clearly state the form of activation functions being used.

(c) An additional GRU is added to the network. Briefly describe the revised network configuration.

4. Layer normalisation (layer norm) is a popular form of normalisation for deep learning. This acts on the the activation function outputs for particular layer. For layer k and input vector $\boldsymbol{x}_p$ the n_k nodes associated with the layer yields the vector of activation function outputs $\boldsymbol{y}_p^{(k)}$. Layer norm is applied to yield the normalised vector $\tilde{\boldsymbol{y}}_p^{(k)}$, which is passed to the next layer, where

$$\tilde{\boldsymbol{y}}_p^{(k)} = \frac{1}{\tilde{\sigma}_p^{(k)}}(\boldsymbol{y}_p^{(k)} - \tilde{\mu}_p^{(k)}\mathbf{1}); \quad \tilde{\mu}_p^{(k)} = \frac{1}{n_k}\sum_{j=1}^{n_k} y_{pj}^{(k)}; \quad \tilde{\sigma}_p^{(k)2} = \frac{1}{n_k}\sum_{j=1}^{n_k}(y_{pj}^{(k)} - \tilde{\mu}_p^{(k)})^2$$

and $\mathbf{1}$ is the vector of ones length n_k.

(a) Compare layer and batch normalisation for training networks. You should consider the form of normalisation applied and any difficulaties/sensitivities in applying the transform during training or inference

(b) How does layer norm alter the derivative $\partial E(\boldsymbol{\theta})/\partial \boldsymbol{y}^{(k)}$ when $\tilde{\sigma}_p^{(k)}$ is ignored in the normalisation (so $\tilde{\sigma}_p^{(k)} = 1$ for all inputs)?

5. (optional) The following form of expression is often used in variational optimisation with training examples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ to find the model parameters $\boldsymbol{\lambda}$

$$\mathcal{L}(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \log(\boldsymbol{x}_i; \boldsymbol{\lambda}) \geq \sum_{i=1}^{n} \int \log\left(\frac{p(\boldsymbol{x}_i, \boldsymbol{z}; \boldsymbol{\lambda})}{q_1(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}_1)}\right) q_2(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}_2) d\boldsymbol{z} = \sum_{i=1}^{n} \left\langle \log\left(\frac{p(\boldsymbol{x}_i, \boldsymbol{z}; \boldsymbol{\lambda})}{q_1(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}_1)}\right) \right\rangle_{q_2(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}_2)}$$

   (a) Show that if $q_1(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}_1) = p(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\lambda})$ for all values of $\boldsymbol{x}$ and $\boldsymbol{z}$ then the left and right-hand sides of this expression are equal for all functions $q_2(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}_2)$.

   (b) Show that if $q_1(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}_1) = q_2(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}_2)$ then equality only occurs when $q_1(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}_1) = p(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\lambda})$ for all values of $\boldsymbol{x}$ and $\boldsymbol{z}$.

   (c) Briefly discuss why this form of expression may be useful.

Support Vector Machines

6. † A binary classifier is to be trained. What are the limitations of linear decision classifiers and why do non-linear mappings of the feature space allow improved discrimination? Under what conditions is it guaranteed that a non-linear mapping will allow perfect classification of the data?

7. In the `XOR` classification from the lectures, if we use the kernel $k(\boldsymbol{x}_n, \boldsymbol{x}_m) = (1 + \boldsymbol{x}_n^{\mathrm{T}} \boldsymbol{x}_m)^2$, what is the resulting Gram matrix? The solution to the dual problem with such kernel is $a_1 = a_2 = a_3 = a_4 = 1/8$. Show that this solution satisfies the constraints $t_n y(\boldsymbol{x}_n) \geq 1$ for $n = 1, \ldots, 4$, where $y(\boldsymbol{x})$ is the output of the classifier for input $\boldsymbol{x} \in \mathbb{R}^2$. What is the equation of the final decision boundary?

8. The following data is to be used for training an SVM

$$\text{Positive class } (t_n = +1): \quad \boldsymbol{x}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{x}_2 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \quad \boldsymbol{x}_3 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}.$$

$$\text{Negative class } (t_n = -1): \quad \boldsymbol{x}_4 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{x}_5 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{x}_6 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

   (a) Plot the training points and, by inspection, draw the optimal, maximum margin, decision boundary.

   (b) What are the support vectors? Let $y(\boldsymbol{x}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{x} + b$ be the classifier's output for input $\boldsymbol{x}$. What are $\mathbf{w}$ and b so that $t_n y(\boldsymbol{x}_s) = 1$ for any support vector $\boldsymbol{x}_s$?

   (c) Express $y(\boldsymbol{x})$ in terms of the Lagrange multipliers, a_n and show that $\mathbf{w}$, b and the a_n satisfy the KKT conditions.

9. $*$ A Support Vector Machine (SVM) is to be used for classifying sequences represented by 1-dimensional feature-vectors of variable length. To solve the classification problem, each input sequence $\boldsymbol{x}_{1:T} = \{x_1, \ldots, x_T\}$ is transformed using the feature mapping

$$\boldsymbol{\phi}(\boldsymbol{x}_{1:T}) = \begin{bmatrix} \frac{\partial}{\partial \mu_1} \log p(\boldsymbol{x}_{1:T}) \\ \vdots \\ \frac{\partial}{\partial \mu_M} \log p(\boldsymbol{x}_{1:T}) \\ \frac{\partial^2}{\partial \mu_1^2} \log p(\boldsymbol{x}_{1:T}) \\ \vdots \\ \frac{\partial^2}{\partial \mu_1 \partial \mu_M} \log p(\boldsymbol{x}_{1:T}) \\ \vdots \\ \frac{\partial^2}{\partial \mu_M^2} \log p(\boldsymbol{x}_{1:T}) \end{bmatrix},$$

where $p(\boldsymbol{x}_{1:T})$ is specified by a generative model given by an M-component Gaussian Mixture Model (GMM):

$$p(\boldsymbol{x}_{1:T}) = \prod_{t=1}^{T} \sum_{m=1}^{M} c_m \mathcal{N}(x_t; \mu_m, \sigma_m^2)$$

(a) Why is this form of feature-space suitable for use with SVMs when classifying variable-length data-sequences? Why is an SVM a suitable form of classifier as M (the number of components) gets large? What is the dimensionality of the feature-space in this case?

(b) Derive an expression for $\frac{\partial}{\partial \mu_i} \log p(\mathbf{x}_{1:T})$. This should be expressed in terms of $P(i|x_t)$, the posterior probability that the i-th Gaussian component generated the observation.

(c) Hence show that

$$\frac{\partial^2}{\partial \mu_j \partial \mu_i} \log p(\boldsymbol{x}_{1:T}) = -\sum_{t=1}^{T} P(i|x_t) P(j|x_t) \frac{(x_t - \mu_j)(x_t - \mu_i)}{\sigma_i^2 \sigma_j^2}.$$

Do you expect these second-order derivative terms to help in classification?

M.J.F. Gales & J. M. Hernandez-Lobato
Oct 2017,2018,2020