Coláiste na Tríonóide, Baile Átha Cliath
Trinity College Dublin
Ollscoil Átha Cliath | The University of Dublin

EE4C16

FACULTY OF ENGINEERING, MATHEMATICS & SCIENCE

SCHOOL OF ENGINEERING

Electronic & Electrical Engineering

Engineering

Senior Sophister

Annual Examinations

Hillary Term, 2017

Machine Learning with Applications in Media Engineering (EE4C16)

Solutions

Dr. F. Pitié

Instructions to candidates:

Answer FOUR (4) questions.

Please answer questions from each section in separate answer books.

Materials Permitted for this Examination:

New Formulae & Statistic Tables

Graph Paper

Non-programmable calculators

Question 1

1. Explain why we need a validation set when developing a Machine Learning system.

[5 marks]

The issue is that training causes overfitting on your training set. In a sense, training "spoils" the set that it uses. We don't want to spoil the test set, as it should gives us a fair estimation of the performance. Hence when we tune for hyperparameters, we need a dedicated set that is not the test set and not the training set. This is the validation set.

2. What does the backpropagation algorithm compute?

[3 marks]

Backpropagation computes the gradient of the loss function of the Neural Network output wrt to the network weights.

3. Explain why backpropagation makes DNN practical.

[5 marks]

Backpropagation is important because gradient descent is the only known effective way of training a network and that computing the gradient in a naive way is excessively expensive. In most typical situations Backpropagation requires $\mathcal{O}(n)$ operations (where n is the number of units), whereas naive gradient approximation using finite differences requires $\mathcal{O}(n^2)$ operations. A modern network easily contains millions of units.

4. Suppose that a credit card company decides to deploy a new AI system for assessing the risk of default of its customers. The new system is using a feed-forward neural network. Suggest, in a form of essay, what should the bank have before the system can be used? Discuss problems associated with this requirement.

[12 marks]

The answer should mention that the company should get hold of historical data about its customers who already took credit in the past. This data will be used as a training set for the neural network. It is important that the data is representative and covers

as many types of customers as possible. This is because the network will not be able to produce an accurate answer for a customer very different from those in the training set

Question 2

1. The estimation of the weights in a DNN is usually based on the gradient descent optimisation technique. Is the gradient descent technique guaranteed to converge to the global minimum? If not, comment on the conditions and the type of convergence.

[5 marks]

No. Gradient descent is only guaranteed to converge for particular choices of learning rates. And then it is only a convergence to a local minimum. Convergence to global minimum can be guaranteed if the problem is convex (eg. Least Squares).

2. "A classifier trained on less data is less likely to overfit". comment.

[4 marks]

That is incorrect. Overfitting happens when the model is rich enough to fit training set. Thus reducing the number of points in the training is actually more likely to cause overfitting. Solutions to overfitting include: 1) increasing the size of the training set and 2) setting regularisation priors.

3. What is a "Batch gradient" descent? What is a "stochastic gradient" descent?

[4 marks]

A "batch gradient" is the gradient averaged over a number of samples ("a batch"). "Stochastic gradient" is the extreme case where the batch size is only 1.

4. Why do RNNs have a tendency to suffer from exploding/vanishing gradient?

[6 marks]

RNN are trained by backpropagation through time. They unfold into deep feed forward net. When gradient is passed back through many time steps, it tends to grow or vanish in same way as it happens in deep feedforward nets.

5. There are a few ways of preventing the problem of vanishing gradients. Name one activation function that can be used to reduce the issue. Name one example of CNN architecture designed to address the issue. Name one example of RNN architecture designed to address the issue.

[6 marks]

We can use ReLU or similar activation functions to reduce the range of values for which the gradient is nearly zero.

For CNN's, we can use Residual networks. Residual networks add direct connections between deep and shallow layers, which has the effect of reducing the number of factors in the gradients computations.

In RNN's, we can use Long Short-Term Memory (LSTM) which is designed to maintain constant the magnitude of the gradient.

Question 3

1. Assuming that the matrix $\mathbf{A}$ is symmetric, and denoting $\mathbf{I}$ as the identity matrix, and λ is real number. Compute the gradient $\frac{\partial E(\mathbf{w})}{\partial \mathbf{w}}$ of $E(\mathbf{w}) = \|(\mathbf{A} + \lambda \mathbf{I})\mathbf{w}\|^2$.

[5 marks]

$$\frac{\partial E}{\partial \mathbf{w}} = 2\left(\mathbf{A} + \lambda \mathbf{I}\right)^2 \mathbf{w}$$

2. We also know that $\mathbf{Bw} \approx \mathbf{b}$. Design a new loss function $E'(\mathbf{w})$ that adapts the loss function $E(\mathbf{w})$ to add a L2 regularisation term to loosely enforce this constraint.

[5 marks]

$$E'(\mathbf{w}) = \|(\mathbf{A} + \lambda \mathbf{I})\mathbf{w}\|^2 + \mu \|\|\mathbf{Bw} - \mathbf{b}\|^2$$

3. Derive the gradient $\frac{\partial E'(\mathbf{w})}{\partial \mathbf{w}}$

[5 marks]

$$\frac{\partial E'}{\partial \mathbf{w}} = 2\left(\mathbf{A} + \lambda \mathbf{I}\right)^2 \mathbf{w} + 2\mu \mathbf{B}^\top \left(\mathbf{Bw} - \mathbf{b}\right)$$

4. Bogus Forensics Ltd. has developed a new test for detecting potential criminals using a new patented hand writing analysis technology. The company claims an 89% accuracy and impresses the investors. Comment.

[5 marks]

Knowing the accuracy alone is pretty much meaningless. At least, you should know about the balance of the test data. Are there any criminal in the dataset? Ideally, they should provide another metric, such as recall or precision.

5. After investigation, you obtain the following confusion matrix for their detector:

	actual: 0	actual: 1
predicted: 0	TN=80	FN=8
predicted: 1	FP=3	TP=9

A positive prediction means that we predict that the person will commit a crime. Make your own analysis on the value of the product.

[5 marks]

The student is expected to talk about the possible imbalance in the data and look at a few metrics.

The number of actual positives is 17 and the number of actual negatives is 83. There is a massive imbalance in the dataset, hence it is no surprise that the accuracy is high.

The accuracy is indeed 89%.

The Recall is 9/(8+9)=53%.

The Precision is 9/(3+9)=75%.

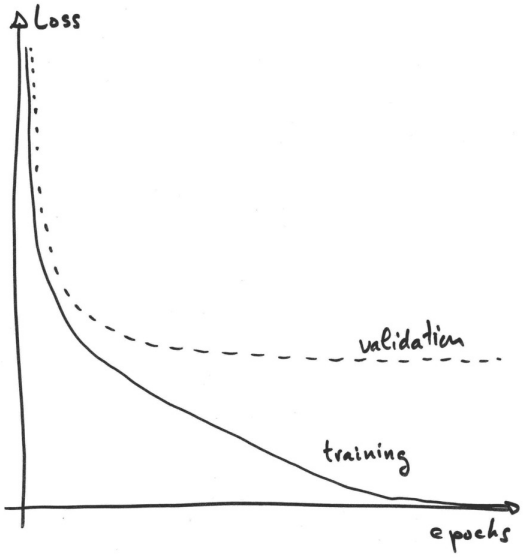
The false positive rate is 3/(3+80)=3.6%.

We note that the Recall at 53% is quite poor.

That said, it is still pretty impressive considering the technology is based on handwriting analysis alone.

Question 4

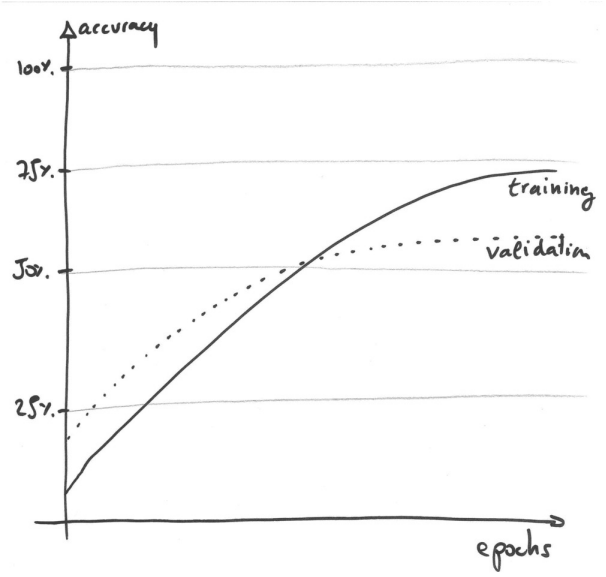
1. While training a convolutional neural net, we obtain a loss function graph similar to the one below. Comment what is happening and explain what your next steps should be.



[5 marks]

The training loss seems to be satisfactory while the loss on the validation dataset fails to go down. This is a classic case of overfitting. The network has enough capacity to learn the learning set but fails to generalise. Several solutions can be implemented: 1) get more data to increase the size of the training set, 2) add or increase regularisation, 3) add or increase Dropout.

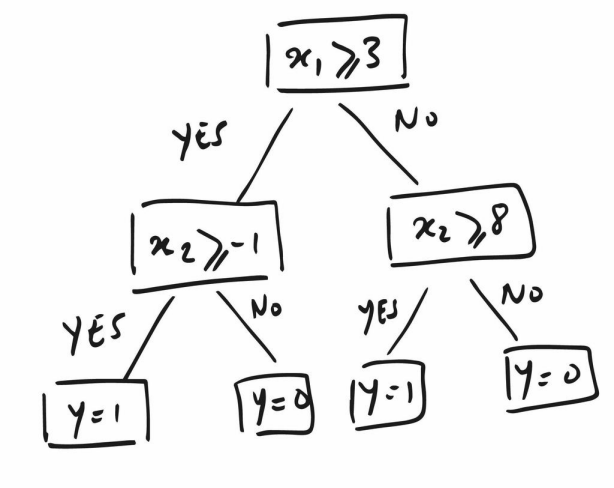
2. While training a convolutional neural net, we obtain an accuracy graph similar to the one below. Comment on the graph and indicate what kind of training technique was probably applied.



[5 marks]

We observe that at the start of the training, the accuracy on the validation set is higher than the accuracy on the training set, and that, as the training progresses the system eventually overfits. This is most likely due to Dropout. With dropout, units are randomly set to 0 during training. This impacts negatively on the accuracy of the DNN during training. After a while, the system has developed some redundancy and the system "learns" how to deal with dropout and, given enough units, the DNN can overfit.

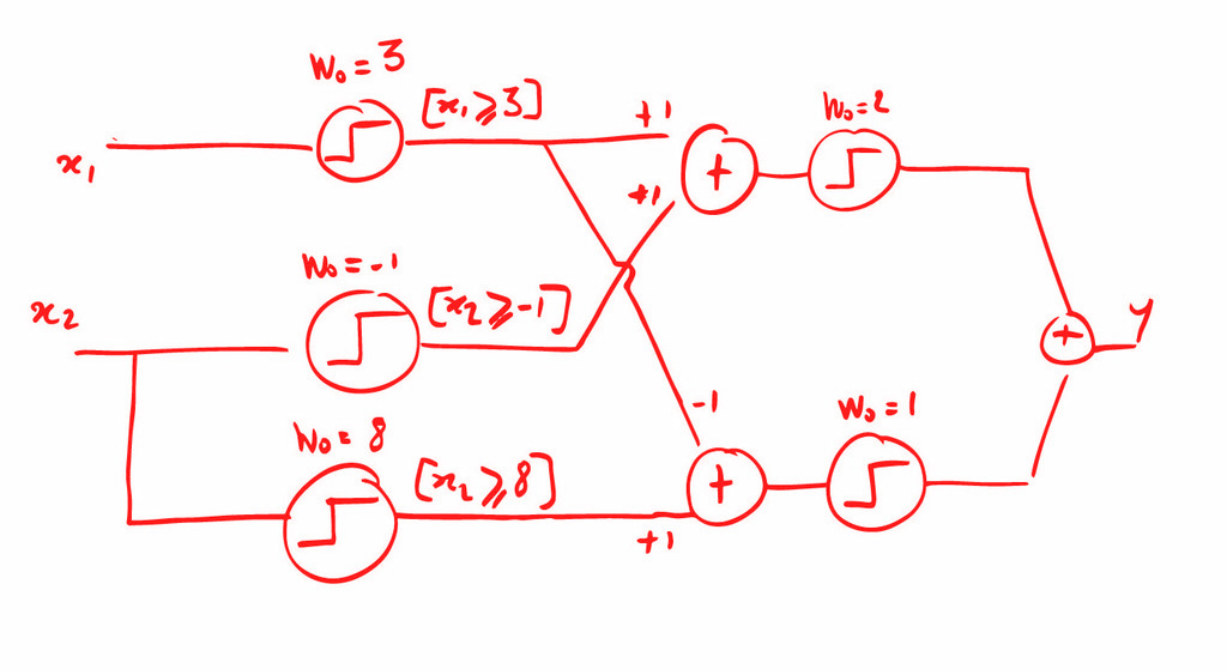
3. Recall what a Decision Tree is and briefly outline why any Decision Tree can be modelled as a neural network. Illustrate this on the simple example below. You can use the binary step activation $f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$.



[15 marks]

A decision tree is a flowchart-like structure in which each internal node represents a binary on an feature, each branch represents the outcome of the test, and each leaf node represents a class label.

The decisions split regions along the axes. The decisions are typically of the form $x_1 \geq T$. These decisions can be modelled by a single neuron with binary step activation where T is the weight. Logical expressions can be derived by adding the results of the individual comparisons.



Question 5

1. Discuss, in the form of an essay, the challenges of deploying DNN applications on embedded devices. The answer should include discussions about the importance of embedded systems in DNN applications, the specific requirements of embedded systems and the different strategies available to optimise the networks to these specific constraints.

[25 marks]

This is an open question that is covered by two of the 4c16 industry keynotes. Here are a few indications.

The essay should stress the importance of *on-the-edge* analysis and the fact that not all processing can possibly happen in the cloud.

The trend is at a slow down of Moore's law. Packing more transistors per device becomes economically questionable.

Thus porting DNNs to embedded devices require consideration w.r.t. power consumption, memory and speed.

The problem is that Current DNN models are optimised for GPU-computational model and need to be adapted before installing on the embedded devices.

This requires a network optimisation stage, which typically include:

(a) Vertical and horizontal Fusion of operations (eg. fusing multiple conv layers or fusing conv layer with pooling layer). The problem is that memory is much slower than cpu. So data Locality is critical. We need to do as much compute as possible on as smallest span of data and keep higher bandwidth data as local as we can. Fusing layers increases locality.

(b) Pruning network for sparsity. (eg. a network that is designed for 100 classes could be pruned if applied to only 2 classes).

(c) Use of lower precision arithmetic. Reducing Precision can help a lot. For instance using 1b precision instead of 32b floating point results in

i. 100x size reduction on the fabric.
ii. 32x memory footprint reduction of the DNN weights
iii. increased speed
iv. reduced energy consumption

Typically the network performance decreases when reducing the arithmetic precision. However this can be compensated by increasing the network size. The increase in size is largely compensated by the benefits presented above.

Profiling DNNs is also a key tool in the prototyping toolchain. We need to know which components of the network take the most amount of time.

Supporting material

Assuming $\mathsf{a}, \mathsf{b}, \mathsf{A}$ are independent of w, below is a list of useful gradient computations:

$$
\begin{aligned}
\frac{\partial \mathsf{a}^\top \mathsf{w}}{\partial \mathsf{w}} &= \mathsf{a} \\
\frac{\partial \mathsf{b}^\top \mathsf{A}\mathsf{w}}{\partial \mathsf{w}} &= \mathsf{A}^\top \mathsf{b} \\
\frac{\partial \mathsf{w}^\top \mathsf{A}\mathsf{w}}{\partial \mathsf{w}} &= (\mathsf{A} + \mathsf{A}^\top)\mathsf{w} \qquad \text{(or } 2\mathsf{A}\mathsf{w} \text{ if } A \text{ symmetric)} \\
\frac{\partial \mathsf{w}^\top \mathsf{w}}{\partial \mathsf{w}} &= 2\mathsf{w} \\
\frac{\partial\, \mathsf{a}^\top \mathsf{w}\mathsf{w}^\top \mathsf{b}}{\partial\, \mathsf{w}} &= (\mathsf{a}\mathsf{b}^\top + \mathsf{b}\mathsf{a}^\top)\mathsf{w}
\end{aligned}
$$