Test 01 – Math

Arthur J. Redfern
arthur.redfern@utdallas.edu
Feb 27, 2019

---

0 Information

Logistics

- Name: _____
- UT Dallas ID: _____

Instructions

- There are 28 numbered questions with indicated point values that sum to 100
- Write all of your answers clearly on this test and turn it in
- No reference materials are allowed
- No help from others is allowed
- Correct answers in red

1 Test

Strategy [11 points]

Consider 4 stages of processing:

Stage 1:	pre processing
Stage 2:	feature extraction
Stage 3:	prediction
Stage 4:	post processing

1. [2 points] List the stages that xNN based methods are commonly used for in this class:

Stage(s): 2 and 3

2. [4 points] List the 2 generic types of prediction problems we've considered in this class:
   2 generic types of prediction problems are: classification and regression

3. [5 points] Circle "true" or false for each of the following statements
   - True / False Neural networks are universal approximators
   - True / False A function always exists that maps an arbitrary input to arbitrary` output
   - True / False A 3 layer neural network is always the best way to approximate a function
   - True / False A small amount of data is always sufficient for good neural network training
   - True / False It's possible to have a deep neural network without nonlinearities

Data [8 points]

Consider an image classification data set $\mathbf{X}$ with 2^{10} classes and 2^{12} labeled examples per class for a total of 2^{22} labeled examples. Let μ be the data set mean and σ^2 be the data set variance.

4. [4 points] What is the total information content of all of the labels?
   Information content of all labels = number of labels * information per label
   $= 2^{22} \log_2(2^{10})$
   $= 10*2^{22}$ bits

5. [4 points] How would you transform the data set $\mathbf{X}$ to a 0 mean unit variance data set $\mathbf{X}_{norm}$?
   $\mathbf{X}_{norm} = (\mathbf{X} - \mu)/\sigma$

Weight initialization [4 points]

6. [4 points] Let's say I know that the value of an individual weight in a network layer is in the range [a, b], but I know nothing else about it. What is the entropy maximizing distribution to sample from to initialize this weight?
   The entropy maximizing distribution is: uniform from [a, b]

Feature extraction – CNN [24 points]

Consider a CNN style 2D convolution layer $\mathbf{Y} = f(\mathbf{H} \otimes \mathbf{X}_{padded} + \mathbf{V})$ where $\otimes$ is used to denote CNN style 2D convolution and

Input:	$\mathbf{X}$ with dimensions $N_i \times L_r \times L_c$
Pad:	P_r (= **sum** of top + bottom pad), P_c (= **sum** of left + right pad)
Filter:	$\mathbf{H}$ with dimensions $N_o \times N_i \times F_r \times F_c$ (no striding)

Bias: $\mathbf{V}$ with dimensions N_o x M_r x M_c and constant per n_o
Nonlinearity: f of type ReLU
Output: $\mathbf{Y}$ with dimensions N_o x M_r x M_c

7. [4 points] What are P_r and P_c such that $M_r = L_r$ and $M_c = L_c$?

$P_r = F_r - 1$
$P_c = F_c - 1$

8. [4 points] What are the matrix dimensions of the resulting CNN style 2D convolution operation $\otimes$ lowered to matrix matrix multiplication expressed using BLAS based M (result rows), N (result cols) and K (operand inner dimension) notation? Assume the pad is chosen as above.

BLAS notation M $= N_o$
BLAS notation N $= L_r * L_c$
BLAS notation K $= N_i * F_r * F_c$

9. [4 points] How many MACs are required in the standard matrix multiplication based implementation of CNN style 2D convolution with the pad chosen as above (note that this does not include the bias and nonlinearity)?

Number of MACs $= L_r * L_c * N_o * N_i * F_r * F_c$

10. [4 points] How many elements of filter memory are in CNN style 2D convolution (note that this does not include the bias and nonlinearity)?

Number of filter memory elements $= N_o * N_i * F_r * F_c$

11. [4 points] How many elements of input feature map + output feature map memory are in CNN style 2D convolution with the pad chosen as above?

Number of input feature map + output feature map elements $= (N_i + N_o) * L_r * L_c$

12. [2 points] Assume that the layer is part of a network and trained for a 3 x 32 x 64 input $\mathbf{X}$. Is the convolution operation mathematically compatible with a 3 x 96 x 96 input $\mathbf{X}$? Circle yes or no.

Yes / No

13. [2 points] Consider the L_r x L_c output feature map at channel n_0. Are the same filter coefficients used for mapping inputs to outputs for all $L_r * L_c$ output pixels in feature map n_o? Circle yes or no.

Yes / No

Feature extraction – RNN [6 points]

Consider a standard RNN layer $\mathbf{y}_t = f(\mathbf{H}\,\mathbf{x}_t + \mathbf{G}\,\mathbf{y}_{t-1} + \mathbf{v})$ with

Input at time t: $\mathbf{x}_t$ with dimensions N_i x 1

Input matrix:	$\mathbf{H}$ with dimensions N_o x N_i
Output at time t–1:	$\mathbf{y}_{t-1}$ with dimensions N_o x 1
State matrix:	$\mathbf{G}$ with dimensions N_o x N_o
Bias:	$\mathbf{v}$ with dimensions N_o x 1
Nonlinearity:	f of type ReLU
Output at time t:	$\mathbf{y}_t$ with dimensions N_o x 1

and the sequential set of inputs $\{\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\}$ and outputs $\{\mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4\}$ with $\mathbf{y}_{-1} = \mathbf{0}$.

14. [2 points] Can all of the input terms $\{\mathbf{H}\ \mathbf{x}_t\}$ for t = 0, …, 4 be computed parallel (at the same time)? Circle yes or no.
Yes / No

15. [2 points] Can all of the state terms $\{\mathbf{G}\ \mathbf{y}_{t-1}\}$ for t = 0, …, 4 be computed parallel (at the same time)? Circle yes or no.
Yes / No

16. [2 points] Assume that there's an error in output $\mathbf{y}_2$. What other output(s) will potentially be in error because of this?
Output(s): $\mathbf{y}_3$, $\mathbf{y}_4$

Down sampling – pooling [8 points]

Consider an input feature map $\mathbf{X}$ of dimension N_i x L_r x L_c where N_i, L_r and L_c are all divisible by 2.

17. [4 points] For a 2x2/2 average pooling layer, what are the dimensions of the output $\mathbf{Y}$?
The dimensions of $\mathbf{Y}$ are N_i x (L_r/2) x (L_c/2)

18. [4 points] For a 3x3/2 max pooling layer, what are the pad value P_r (= **sum** of top + bottom pad) and P_c (= **sum** of left + right pad) that result in an output feature map $\mathbf{Y}$ with dimensions N_i x (L_r/2) x (L_c/2)?
P_r = 1 or 2
P_c = 1 or 2

Nonlinearity choices [3 points]

Consider the following statements:

Statement A:	zeros out negatively aligned features, does not change positively aligned features
Statement B:	constrains the output to (-1, 1)
Statement C:	constrains the output to (0, 1) and is frequently used as a gate

19. [3 points] Match statements A, B and C to the corresponding nonlinearity

ReLU → Statement A
Sigmoid → Statement C
Tanh → Statement B

Prediction [6 points]

Consider a network designed for image classification with

Input $\mathbf{X}_0$ with dimensions N_i x L_r x L_c
Multiple CNN and pooling layers
Feature map $\mathbf{X}_d$ with dimensions N_d x (L_r/D) x (L_c/D)
Global average pooling layer
Dense layer with no nonlinearity
Output $\mathbf{y}$ with dimensions classes x 1

20. [6 points] Circle "true" or false for each of the following statements

True / False The global average pooling layer allows the dense layer to be mathematically compatible with feature map $\mathbf{X}_d$ given input $\mathbf{X}_0$ with ~arbitrary rows and cols

True / False The purpose of all the layers before the dense layer is to transform the input $\mathbf{X}_0$ into linearly (affine) separable classes

True / False All of the elements in output $\mathbf{y}$ should be similar in value for a properly designed and functioning network

Error computation [8 points]

Consider the above network designed for image classification and a 1 hot training vector $\mathbf{p}^*$ with a 1 in the position corresponding to the class of input $\mathbf{X}_0$.

21. [2 points] What is the name of the function used to transform network output $\mathbf{y}$ to probability mass function $\mathbf{p}$?

$\mathbf{p}$ = Softmax($\mathbf{y}$)

22. [2 points] What is the name of the function commonly used to compute error e from the network predicted probability mass function $\mathbf{p}$ and the true probability mass function $\mathbf{p}^*$?

e = CrossEntropy($\mathbf{p}^*$, $\mathbf{p}$)

23. [2 points] Circle "true" or false for each of the following statements

True / False For numerical stability in the gradient calculation, these 2 functions are commonly included as a single function in high level neural network software libraries

Consider a regression network designed to compute an output vector **y** and a training vector **y*** representing the ideal vector.

24. [2 points] What is the name of the function commonly used to compute error e from the network predicted vector **y** and the ideal vector **y***?

e = MeanSquareError(**y***, **y**)

Back propagation [10 points]

25. [4 points] Circle "true" or false for each of the following statements

True / False A graph for back propagation can be constructed from the graph for forward propagation

True / False For end to end training with back propagation, it's ok if a few layers are not differentiable or sub differentiable.

Consider a residual building block with input **x** and output **y** = **x** + **f**(**x**) and assume that ∂e/∂**y**, the sensitivity of the error e with respect to feature map **y**, is known.

26. [6 points] Write ∂e/∂**x** in terms of ∂**f**/∂**x** and ∂e/∂**y**.

Identity path gradient: (∂e/∂**y**)

Residual path gradient: (∂**f**/∂**x**) (∂e/∂**y**)

∂e/∂**x** = (∂e/∂**y**) + (∂**f**/∂**x**) (∂e/∂**y**)

Weight update [12 points]

Given:

A is symmetric positive definite

α is a scalar

Operator ∂/∂(**h** – $\mathbf{h}_0$) applied to e($\mathbf{h}_0$) = **0**

Operator ∂/∂(**h** – $\mathbf{h}_0$) applied to (**h** – $\mathbf{h}_0$)^T **g** = **g**

Operator ∂/∂(**h** – $\mathbf{h}_0$) applied to 0.5 (**h** – $\mathbf{h}_0$)^T **A** (**h** – $\mathbf{h}_0$) = **A** (**h** – $\mathbf{h}_0$)

27. [6 points] Let error e(**h**) = e($\mathbf{h}_0$) + (**h** – $\mathbf{h}_0$)^T **g** + 0.5 (**h** – $\mathbf{h}_0$)^T **A** (**h** – $\mathbf{h}_0$). What is the optimal choice of **h** – $\mathbf{h}_0$ to minimize the error?

∂e/∂(**h** – $\mathbf{h}_0$) = **0** + **g** + **A** (**h** – $\mathbf{h}_0$)

= **0**

h – $\mathbf{h}_0$ = – $\mathbf{A}^{-1}$ **g**

28. [6 points] Now force $\mathbf{h} - \mathbf{h}_0 = -\alpha\,\mathbf{g}$ such that error $e(\mathbf{h}) = e(\mathbf{h}_0) - \alpha\,\mathbf{g}^T\mathbf{g} + 0.5\,\alpha^2\,\mathbf{g}^T\mathbf{A}\,\mathbf{g}$. What is the optimal choice of α to minimize the error?

$\partial e/\partial\alpha = -\mathbf{g}^T\mathbf{g} + \alpha\,\mathbf{g}^T\mathbf{A}\,\mathbf{g}$

$= 0$

$\alpha = (\mathbf{g}^T\mathbf{g}) / (\mathbf{g}^T\mathbf{A}\,\mathbf{g})$