MSE | MASTER OF SCIENCE IN ENGINEERING

Typical Exam Questions

TSM_DeLearn

Some are rather sketchy

Jean Hennebert
Martin Melchior

Type of Questions

- **Targeting theory and concept:** Focus on definitions given in class (blue frames), understanding concepts such as loss function, gradient descent, optimisation/ regularisation procedures, computational graphs, “static” graph strategy used by TensorFlow, functioning of layers composing a CNN, intuition behind deep architectures, etc.

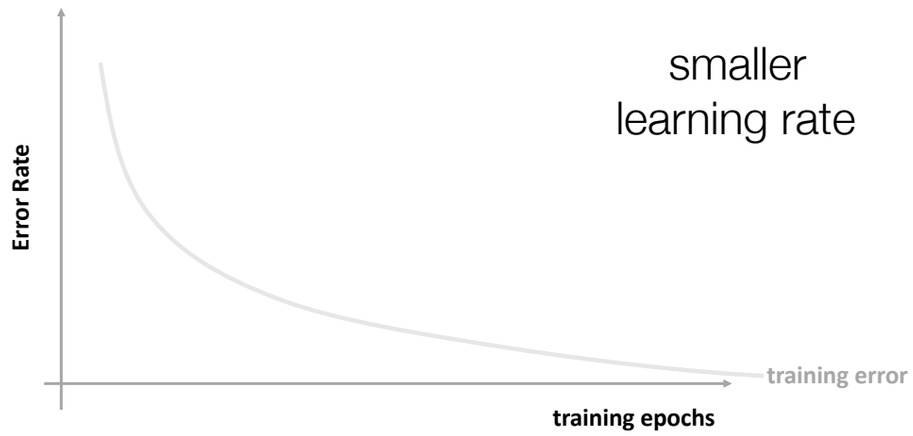
- **Pen-paper exercise:** E.g. computing the numerical values of gradients in a computational graph, computing the activation map on a simple example, computing the number of parameters of a given architecture.

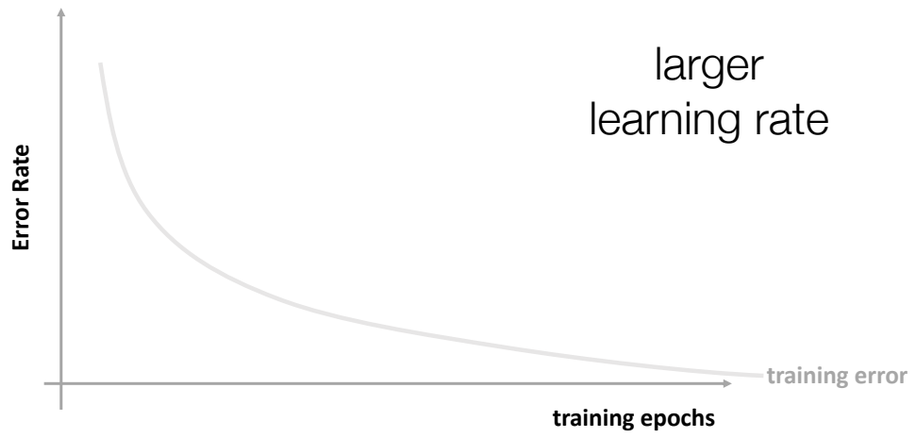
- **Code reading:** Re-explaining a piece of code that you had to produce during the practical works, eventually filling holes in a proposed code (we will not penalise syntax error, we want to see concept such as: here I would add a Dense layer for ex)

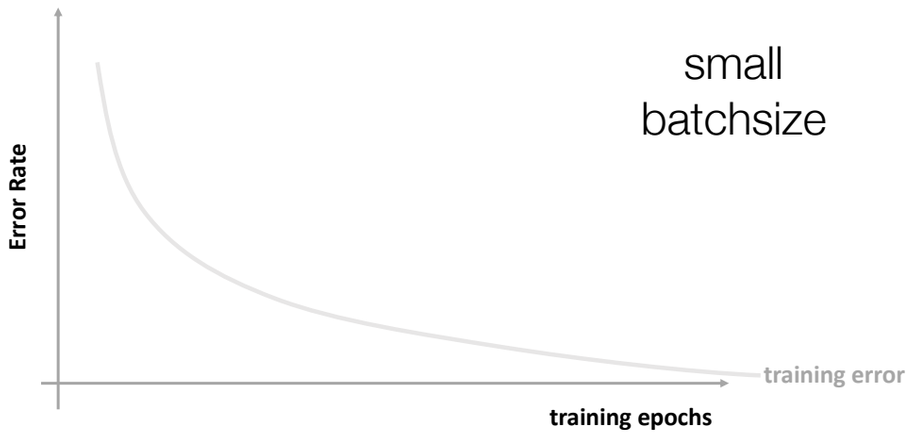
- **Use cases questions:** In front of a given real-life problem described in, let’s say half a page, what would be your deep learning strategy

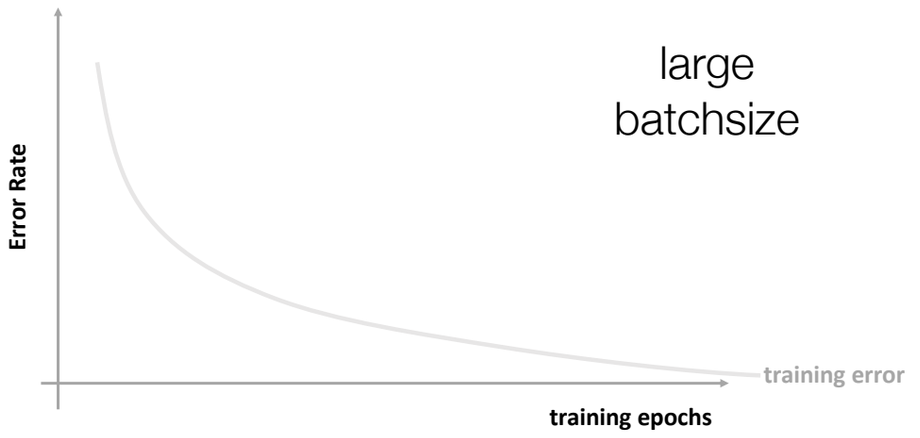
Example Questions: Learning Curves (1)

- Plot qualitatively the changes you expect when modifying the indicated hyper-parameter in the given direction. Express that in 1-2 sentences and explain why it changes the way you expect. The given grey line provides a smoothed baseline.



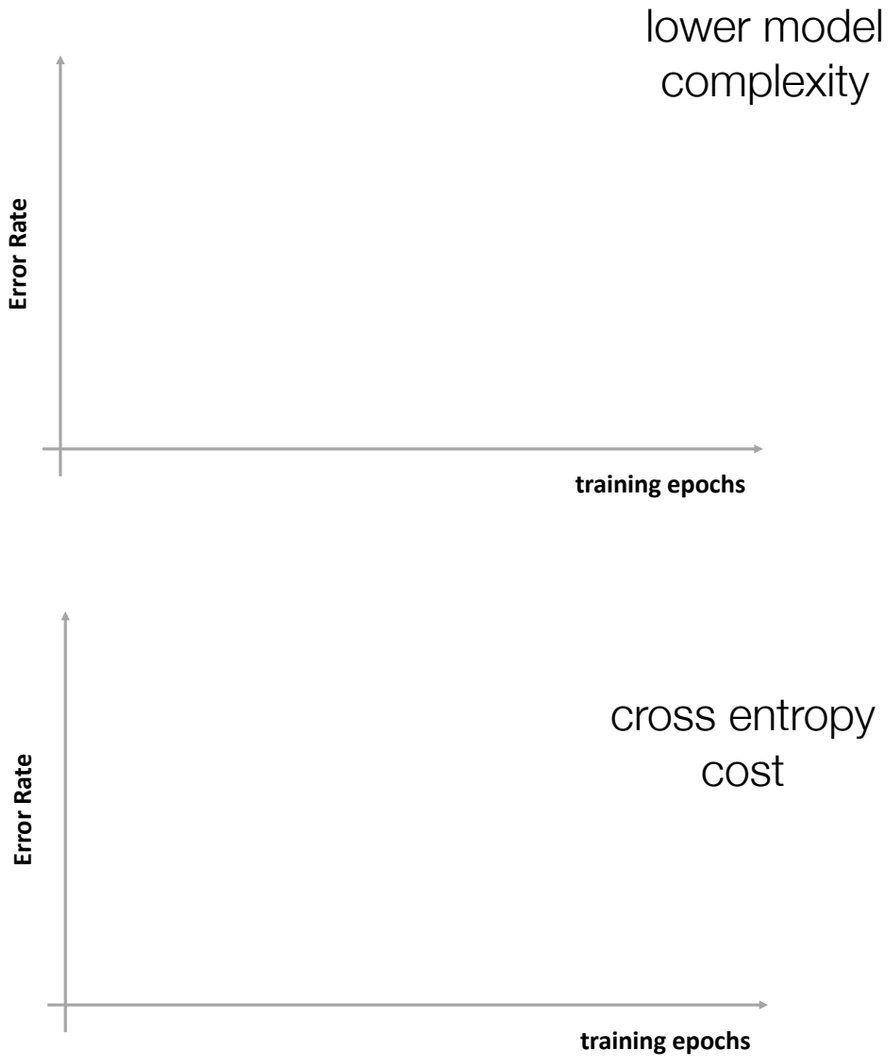


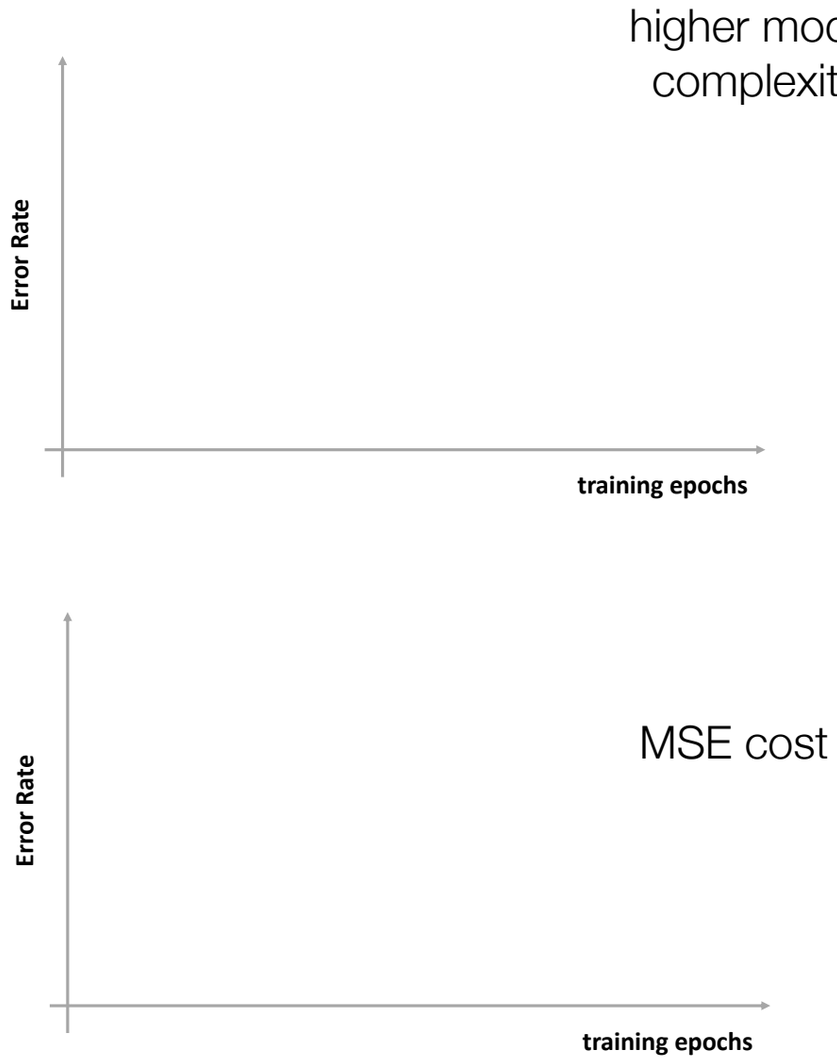




Example Questions: Learning Curves (2)

- Plot qualitatively the train and test error rates you expect for the given change in the settings (to be compared horizontally). Express that in 1-2 sentences and explain why it changes the way you expect.

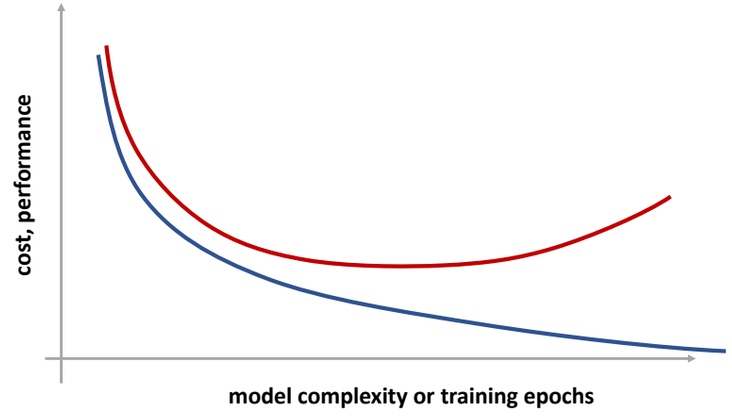




- What change do you expect to see when the parameters (weights) are not properly initialised?

Example Questions: Learning Curves (3)

- Identify overfitting/underfitting from given learning curves
  - In the plots identify the quantities: variance error, bias and generalisation error, test error, training error.
  - Describe the means to make them smaller.

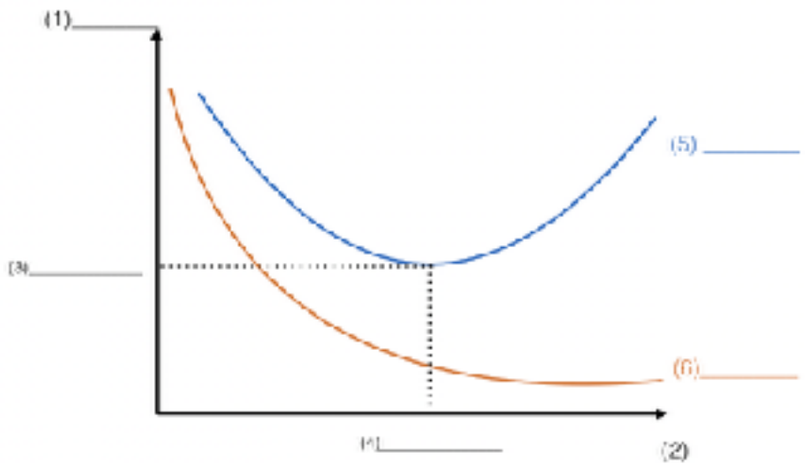


- Explain what effect will the operations listed in the table generally have on the bias and variance of your model. Fill in one of ‘increases’, ‘decreases’ or ‘no change’ in each of the cells.

	Bias	Variance
Regularizing the weights		
Increasing the size of the layers (more hidden units per layer)		
Using dropout to train a deep neural network		
Getting more training data (from the same distribution as before)		

- Explain how the training and the validation/test cost change with increasing number of samples.

- Explain the concept of early stopping with the figure on the r.h.s. Fill-in the blanks: (1) and (2) describe the axes, (3) and (4) describe values on the vertical and horizontal axis, (5) and (6) describe the curves. Be precise.



Example Questions: Update/Optimisation

- Cost Function
  - Explain what the cost function is used for. Specify (incl. formulas) the typical costs for a regression task or a classification task.
  - Formulate the explicit update step for a linear regression problem. Then add L2 regularisation and compute the modified update rule. What will be effect of this regularisation? [Here, we probably would be a bit more specific]
  - Explain why cross entropy is better suited for classification tasks.
  - Compute the cross entropy loss for given labels y=1,3,2 and model scores $\hat{\mathbf{y}} = \begin{pmatrix} 0.4 \\ 0.3 \\ 0.3 \end{pmatrix}, \begin{pmatrix} 0.3 \\ 0.5 \\ 0.2 \end{pmatrix}, \begin{pmatrix} 0.3 \\ 0.5 \\ 0.2 \end{pmatrix}$

- Gradient Descent
  - Describe applicability and convergence properties of gradient descent
  - Explain why gradient descent is considered as a ‘local’ learning principle and what the implications of this are.
  - Describe characteristics and pro’s/con’s of batch gradient descent, mini-batch gradient descent, stochastic gradient descent.

- Mini-Batch Gradient Descent
  - Explain why mini-batch gradient descent is considered as preferred over batch or stochastic gradient descent.
  - Write down pseudo-code for mini-batch gradient descent for given
    - model function `model(x,p)` with input data **`x`**, parameters **`p.`**
    - function for the gradient of the per sample loss w.r.t. the parameters **`p, gradient(p,x)`**
    - input dataset provided by a 2d numpy array with shape **`(n,m)`** with **`n`** input features and **`m`** samples.

Example Questions: General Machine Learning

- Draw a flow chart for a typical machine learning pipeline. Give a short description for each of the steps (1 sentence each) and explain in what situation (1 sentence each) and why each of the steps is important (1 sentence each).

- What's the risk with tuning hyper-parameters using a test dataset?

- Assume you design a model to predict the presence or absence of a tumor in a brain scan. The goal is to ultimately deploy the model to help doctors in hospitals.
  - What performance metrics would you prefer — recall or precision? Explain.

- Consider the confusion matrix

	Apred	Bpred	Cpred
A	20	2	3
B	4	10	1
C	3	2	15

  - Compute the total number of samples. Are these samples of the training or test dataset?
  - Write down the confusion table for class A
  - Compute the overall accuracy and overall error rate of the system
  - Compute the per class recall for all three classes
  - Compute the per class precision for all three classes

Example Questions: Representational Capacity

- Given several models for an image classification task (model complexity indicated) and given error rates: Indicate what error rates are produced by which model. Explain!

- Explain why non-linear activation functions are an important ingredient of neural networks.
- Describe pro's and con's of using relu vs sigmoid. Explain possible issues one needs to deal with when learning models with either of the two functions. Provide typical situations where the two are applied.

- Sketch a constructive proof for why shallow networks (with a single hidden layer and sigmoid activation function) can be used to arbitrarily well approximate continuous functions on finite intervals.
- Explain what the "universal approximation theorem" implies in view of using shallow networks. Explain the difficulties that occur when using only shallow networks and explain in what sense deep networks provide help.

- Consider an neuron with sigmoid activation function and scalar input x, weight w and bias b, i.e. $\sigma(wx + b)$. When compared with w=1, b=0 (i.e. $\sigma(x)$) describe the impact of choosing
  - a negative bias
  - a large positive weight (e.g. w=10)
  - a small negative weight (e.g. w=-0.5)

Example Questions: Multi-Layer Perceptron (1)

- Compute the number of trainable parameters for a MLP with layer sizes 100,50,20 that processes images of size 20x20x3.
- Given a binary classification task for images of shape 64x64x3. The task is modelled with a network with a single hidden layer of 100 units. Compute how much RAM is needed to represent the trainable model parameters in memory (assume that each parameter is represented by float values of 64 bits.

- Compute the softmax values for the given matrix of logits (samples in the columns). Express the result with exponentials without summation symbol.

$$\mathbf{z} = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 3 & 1 \\ 2 & 2 & 0 \end{pmatrix}$$

- Suppose you have a 3-dimensional input x= (x1,x2,x3)=(2,2,1) fully connected to 1 neuron with activation function gi. The forward propagation can be written:

$$z = \sum_{k=1}^{3} w_k x_k + b \ , \ a_i = g_i(z)$$

  After training this network, the values of the weights and bias are: w=(0.5,-0.1,0) and b=0.2. You try 4 different activation functions (g1,g2,g3,g4) which respectively output the values (a1,a2,a3,a4). What is a valid guess for the activation functions gi ?

$$gi \in \{\text{sigmoid}, \text{relu}, \text{heaviside}, \text{linear}\}$$

- Why do we need a bias?
- Explain why we need to randomly initialise the weights. Explain why the scale of the initial weights needs to be chosen carefully.

Example Questions: Multi-Layer Perceptron (2)

- Consider an input image of shape 500×500×3. You flatten this image and use a fully connected layer with 100 hidden units. What are the shapes of the weight matrix and the bias vector?
- Consider an input image of shape 500×500×3. You run this image in a convolutional layer with 10 filters, of kernel size 5×5. How many parameters does this layer have?

- Fill in the blanks — numpy code for computing a batch-normalised relu layer. Also specify the shape (marked with red circles)

```
def batchnorm_layer(X, W, b, gamma, beta):
    '''
    Computes the functionality of batch normalisation with (in the given order)
    1) an affine transformation
    2) relu activation function
    3) batch normalisation
    The layer has

    Arguments:
    X - numpy array of shape (nL, batchsize)
    W - weights matrix of shape (?,?)
    b - bias vector
    gamma - scale parameters of shape (?,?)
    beta - shift parameters of shape (?,?)

    Returns:
    A - activations of layer for given mini-batch of shape (?,?)
    '''

    # Hyper Parameters
    # eps - to avoid division by zero, for numerical stability
    eps = 10^(-8)

    ### START CODE HERE ###









    ### END CODE HERE ###
    return A
```

Example Questions: Regularisation

- Modify the mean-square error loss with a term to accomplish L2 regularisation. Given the formula for the gradient descent update rule. How is it modified w.r.t to the update rule without regularisation?
- Why do we often refer to L2-regularization as “weight decay”? Derive a mathematical expression to explain your point.

- Explain the steps to be performed (at training and test time) when implementing dropout and explain why dropout in a neural network acts as a regulariser.

- Give a method to fight exploding gradient in fully-connected neural networks.

- Assume we have a layer with n=3 units and relu activation function. Assume that we have a mini-batch with m=4 samples. The logit values are given by the matrix

$$\mathbf{z} = \begin{pmatrix} 12 & 14 & 14 & 12 \\ 0 & 10 & 10 & 0 \\ -5 & 5 & 5 & -5 \end{pmatrix}$$

  - Compute the normalised z-values (z_{norm}) and the activations of the layer (expressed as a matrix) by assuming the scale and shift parameters introduced by batch norm to be 1 or 0, respectively (apply batch norm on z before applying ‘relu’).
  - How many parameters (for scale and shift) are added by batch normalisation?
  - Give two benefits of using a batch normalisation.
  - Describe how batch-normalisation is handled at test or production time.

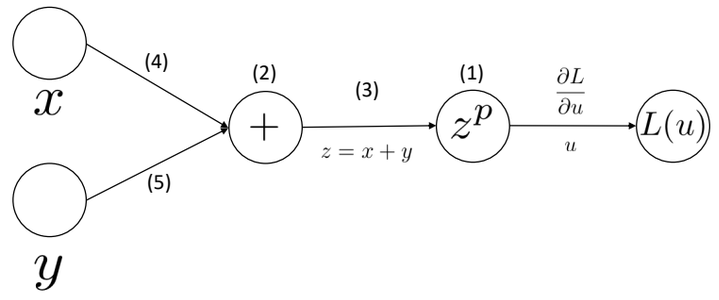
Example Questions: Comp Graphs, Backprop

- Draw the computational graph for the function.

$$f(x, w, b) = 1 + \frac{1}{1 + (w \cdot x + b)^2}$$

At the nodes only the operations +,-,*,/ and powers (.)p are permitted.

- For the computational graph depicted below, specify the local gradients at the nodes (1), (2) and the gradients on the edges (3)-(5) - the latter by using $\frac{\partial L}{\partial u}$



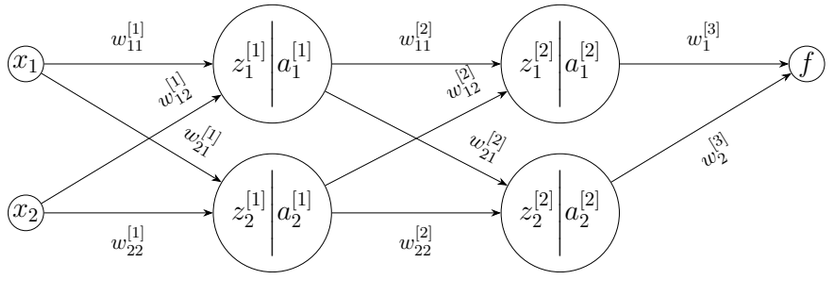
- Compute the local gradient for a max pooling layer and given input into the pooling layer [to be further specified].

- For the network with two hidden layers as depicted on the r.h.s. and with

$$f = w_1^{[3]} a_1^{[2]} + w_2^{[3]} a_2^{[2]}$$

compute the derivatives

$$\frac{\partial f}{\partial z_1^{[2]}} \text{ and } \frac{\partial f}{\partial w_2^{[2]}}$$



$$Z^{[1]} = \begin{bmatrix} z_1^{[1]} \\ z_2^{[1]} \end{bmatrix} = \begin{bmatrix} w_{11}^{[1]} & w_{12}^{[1]} \\ w_{21}^{[1]} & w_{22}^{[1]} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad , \quad A^{[1]} = \begin{bmatrix} a_1^{[1]} \\ a_2^{[1]} \end{bmatrix} = \begin{bmatrix} \sigma(z_1^{[1]}) \\ \sigma(z_2^{[1]}) \end{bmatrix}$$

$$Z^{[2]} = \begin{bmatrix} z_1^{[2]} \\ z_2^{[2]} \end{bmatrix} = \begin{bmatrix} w_{11}^{[2]} & w_{12}^{[2]} \\ w_{21}^{[2]} & w_{22}^{[2]} \end{bmatrix} \begin{bmatrix} a_1^{[1]} \\ a_2^{[1]} \end{bmatrix} \quad , \quad A^{[2]} = \begin{bmatrix} a_1^{[2]} \\ a_2^{[2]} \end{bmatrix} = \begin{bmatrix} \sigma(z_1^{[2]}) \\ \sigma(z_2^{[2]}) \end{bmatrix}$$

Example Questions: CNNs (1)

- Consider a convolutional layer with k=10 filters of f=width=height=5, padding p=0 and stride s=2, ‘relu’ activation
  - Compute the shape of the output if the input has shape 28x28x3.
  - Compute the number of parameters that need to be trained.

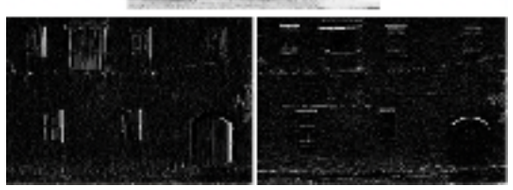
- For input shape 28x28x3 compute the shape of the output for the following network:
  - k=10 filters with f=width=height=5, padding p=1 and stride s=2, ‘relu’ activation function
  - max pooling with stride s=2
  - k=15 filters with f=width=height=3, padding p=2 and stride s=2, ‘relu’ activation
  - 20 filters with width=height=1, padding p=0 and stride s=1, ‘relu’ activation

- What padding needs to be used for a convolutional layer with 3 filters of f=height=width=5 and stride s=1 applied to input images of shape 28x28x3 so that the output and input shape are the same?

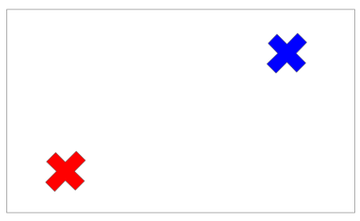
- Compute the result of applying a filter followed by a max pooling. What information needs be cached so that it can be used during back-propagation?

Example Questions: CNNs (2)

- Which of the following is true about max-pooling?
  - It allows a neuron in a network to have information about features in a larger part of the image, compared to a neuron at the same depth in a network without max pooling.
  - It increases the number of parameters when compared to a similar network without max pooling.
  - It increases the sensitivity of the network towards the position of features within an image.
- Look at the grayscale image at the top of the collection of images on the right.
  - Deduce what type of convolutional filter was used to get each of the lower images.
  - Explain briefly and include typical values of these filters. The filters have a shape of (3,3).



- Hand-engineer a filter to be convolved over the image (single channel) on the left that provides the position of the 3x3 black cross.
- Hand-engineer a filter to be convolved over the image (three channels, rgb) on the right that provides the position of the 3x3 red cross but not the blue cross.



Example Questions: CNNs (3)

- Describe as precisely as possible what the filters depicted below are designed to detect.:

a)

0	1	0	0
0	1	0	0
0	1	0	0
0	1	0	0

b)

0	0	0	1
0	0	1	0
0	1	0	0
1	0	0	0

c)

1	1	1	1
1	0	0	0
1	0	0	0
0	0	0	0

- Specify a 3x3 filter good for detecting horizontal edges.

- Code Example Keras (…)