NAME OF THE STUDENT	

DEEP LEARNING FOR ARTIFICIAL INTELLIGENCE — Final exam (21/01/2019) - solution

Master's degree in Telecommunications Engineering (MET), ETSETB TelecomBCN, UPC — Coordinator: Xavier Giró

GENERAL INSTRUCTIONS

This exam follows the general ETSETB regulations.

SPECIFIC INSTRUCTIONS

-
- Write down now your name in the available header of all pages. Unnamed pages will be ignored.
- Time available: 100 minutes.
- Maximum grade is 10 points, and each question has a weight of 1 point.
- Answer in the provided sheets of paper, within the available space. Sheets are structured by instructor to facilitate grading.
- You can only use writing material. No other information sources are allowed in this test.
- If you have any question, raise your hand and wait for instructions from the instructor.
- Any attempt of fraud will be persecuted and punished according to the school regulations.

REVIEW & PUBLICATION OF GRADES

- Publication of preliminary grades: Thursday 24th January @ Atenea.
- Access to the corrected exam: Friday 25th January, 6pm @ D5-003.
- Deadline for the submission of appeals and queries: Monday 28th January @ ETSETB Intranet.
- Publication date of reviewed grades: Tuesday 29th January @ Atenea.

D6L2 INCREMENTAL LEARNING by Ramon Morros

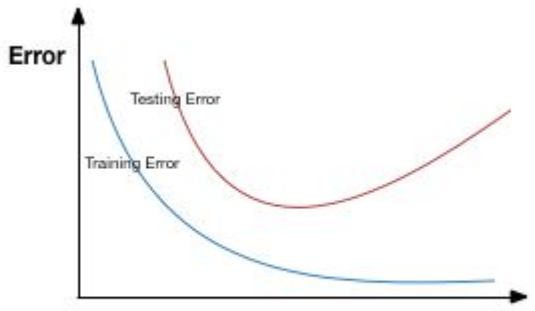
Describe the differences between *multitask learning*, *transfer learning* and *incremental learning*.

Solution:

- In multitask learning, multiple tasks are learned simultaneously. All tasks have labelled data and are treated equally. The training data for all tasks must be available at the moment of training. The goal is to optimize learning/performance across all tasks through shared knowledge.
- In Transfer learning, data in the source domain helps learning the target domain. Less data is needed to train in the target domain. The goal is to optimize performance in the target domain. Transference of knowledge is unidirectional: from source to target domain. This usually results in reduced performance on the source domain (catastrophic forgetting)
- Incremental learning is a form of transfer learning where new data (domain shift, new classes) can arrive sequentially. The performance in the original domain is preserved without having to keep the source domain training data. The goal is optimize the performance on both source and target domains. The target domain usually has less (or not at all) disponibility of supervised labels.

D7L2 METHODOLOGY by Javier Ruiz

"Early stopping" is one technique to prevent overfitting when training deep networks.The following graph shows the training and validation/testing errors by training steps. Answer the following questions:



a) Explain the "early stopping" mechanism and justify when would you stop if your training has a similar graph as the one on the left.

b) Enumerate another 3 possible solutions or algorithms to prevent overfitting in your network.

c) Given the possibility, would you use a higher capacity network and prevent overfitting or a lower capacity one and

prevent underfitting?

Solution:

a) Stop your training when validation error increases. In the graph we would stop our training when the red curve starts to go up.
b) Loss regularization, Data augmentation, Dropout, Parameter sharing, Adversarial training among others.
c) It is usually better to use a higher capacity network and prevent overfitting

D7L1 GENERATIVE MODELS: VAE by Santiago Pascual

A variational autoencoder is a deep generative model with an encoder-decoder architecture, whose loss function is defined as:

$$\log P(X) - D_{KL}[Q(z|X)\|P(z|X)] = E[\log P(X|z)] - D_{KL}[Q(z|X)\|P(z)]$$

Answer the following questions:

(a) What part of the neural network is the term Q(z|X)? and what part is P(X|z)?
(b) Why an autoencoder (NOT variational) is not a generative model?
(c) Why do we normally choose P(z) to be a Normal distribution z ~ N(0, I) ?

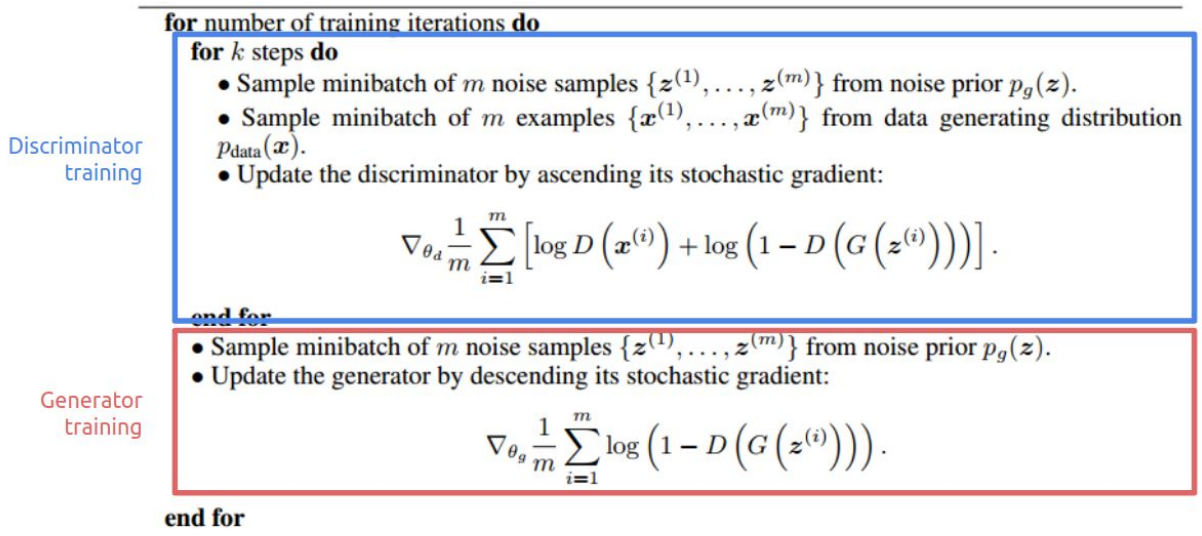
Solution:

(a) Q(z\|X) = encoder, P(X\|z) = decoder
(b) Because it just creates deterministic codes, it can't generate novel samples, no stochasticity, just memorization.
(c) Because of reparameterization trick to make backpropagation work.

D9L1 GENERATIVE MODELS: GANs by Santiago Pascual

In a generative adversarial network (GAN) there are 3 steps within each training iteration (mini-batch) to update both G and D:
(1) D is updated to classify **x** data points from the trainset as real (D(**x**) = 1).
(2) D is updated to classify ẋ = G(**z**) generated data points as fake (D(G(**z**)) = 0).
(3) G is updated to make D(G(**z**)) misclassify its samples as real while D weights are frozen (D(G(**z**)) ~ 1).
In the original "generative adversarial nets" paper by I. Goodfellow et al. they proposed the use of a hyperparameter K during the GAN training that made steps (1) and (2) happen K times prior to going to step (3) in a mini-batch update, as shown in the pseudo-code below:

NAME OF THE STUDENT	



Explain why it might be positive for the learning of the generator to update the discriminator K times per each generator update, with K > 1.

Solution:

We want a good discriminator so that its features to capture what is real or not are useful during backpropagation for the faster learning of the generator. We update the discriminator more often to make it more accurate.

D9L2 GENERATIVE MODELS: LIKELIHOOD MODELS by Santiago Pascual

WaveNet is an autoregressive deep generative model, which means that for a stream of samples **x** = {x[1], x[2], x[3], …, x[T] } composing the datapoint **x** of T dimensions, each predicted sample depends only on the past (x[t] depends on x[t-1], x[t-2], etc.) . Answer the following questions:

(a) Why does this model need to be autoregressive to represent our data likelihood p(**x**)?
(b) How do we make one-dimensional convolutions causal so that they only depend on past information to predict next sample?
(c) What is a dilated convolution and why is it useful in WaveNet?
(d) What is the main advantage of an explicit likelihood model like WaveNet over GANs?

Solution:

(a) Because of probability chain rule, factorizing the join distribution p(x) into the product of conditionals p(x) = prod_{t=1}^{T} p(x[t] | x[1], …, x[t - 1]) .
(b) Just add (kernel_width - 1) zeros on the left side of the input signal.
(c) A convolution with blanks in between kernel weights. It allows for a larger receptive field, thus models a higher dimensional distribution.
(d) The evaluation. Likelihood-based models give us a clear metric that tells us how good our model is fitting the training by direct calculation of p(x), whereas GANs don't.

NAME OF THE STUDENT	

DEEP LEARNING FOR ARTIFICIAL INTELLIGENCE — Final exam (21/01/2019) - solution

Master's degree in Telecommunications Engineering (MET), ETSETB TelecomBCN, UPC — Coordinator: Xavier Giró

D8L1 RECURRENT NEURAL NETWORKS I by Marta R. Costa-jussà

(A) What is one key difference between the backpropagation algorithm and the backpropagation through time algorithm?

Option 1. Backpropagation through time sums up gradients for corresponding weight for each time step and backpropagation does not do it.

Option 2. Unlike backpropagation, Backpropagation through time does not take into account gradients for corresponding weight for each time step

(B) Why are vanishing gradients a more common problem in basic RNNs compared to feed forward networks?

(C) Mention at least 2 ways of facing the vanishing gradient problem.

Solution:

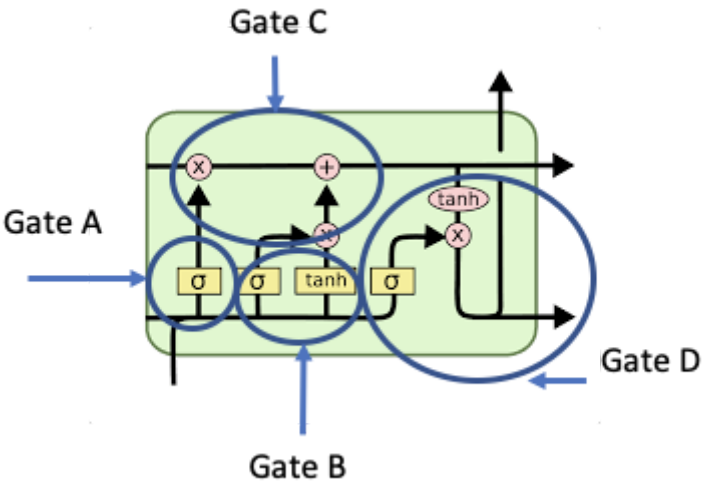
(A) Solution Option 1. BPTT is used in context of recurrent neural networks. It works by summing up gradients for each time step

(B) Vanishing gradients appear when training convergence with gradient descent takes a long time. RNNs include context (hidden units) in their training, while FFNs do not.

(C) Using gated units, using activation functions such as RELU

D8L2 RECURRENT NEURAL NETWORKS II by Marta R. Costa-jussà

(A) Mark each gate name of the picture:



(B) Explain (with examples) the function of each gate.

Solution: (A) Gate A= Forget Gate; Gate B= Input Gate; Gate C= Update Gate; Gate D= Output Gate

(B)

Forget gate: In language model, the cell state might include the gender of the present subject, so that the correct pronouns can be used. When we see a new subject, we want to forget the gender of the old subject.

Input Gate: In the example of our language model, we'd want to add the gender of the new subject to the cell state, to replace the old one we're forgetting.

Update Gate: In the case of the language model, this is where we'd actually drop the information about the old subject's gender and add the new information, as we decided in the previous steps.

NAME OF THE STUDENT	

Output Gate: For the language model example, since it just saw a subject, it might want to output information relevant to a verb, in case that's what is coming next. For example, it might output whether the subject is singular or plural, so that we know what form a verb should be conjugated into if that's what follows next.

D10L1 ATTENTION-BASED MODELS I by Marta R. Costa-jussà

Decide if the next statements about attention-based models are true or false, and elaborate on your responses.

(A) With an attention mechanism we no longer try encode the full source input into a fixed-length vector. Rather, we allow the decoder to "attend" to different parts of the input at each step of the output generation
(B) Attention has only successfully been applied to text sequences. Other applications such as image or speech have tested attention-based mechanisms but without success.

Correctly Complete the following statements:

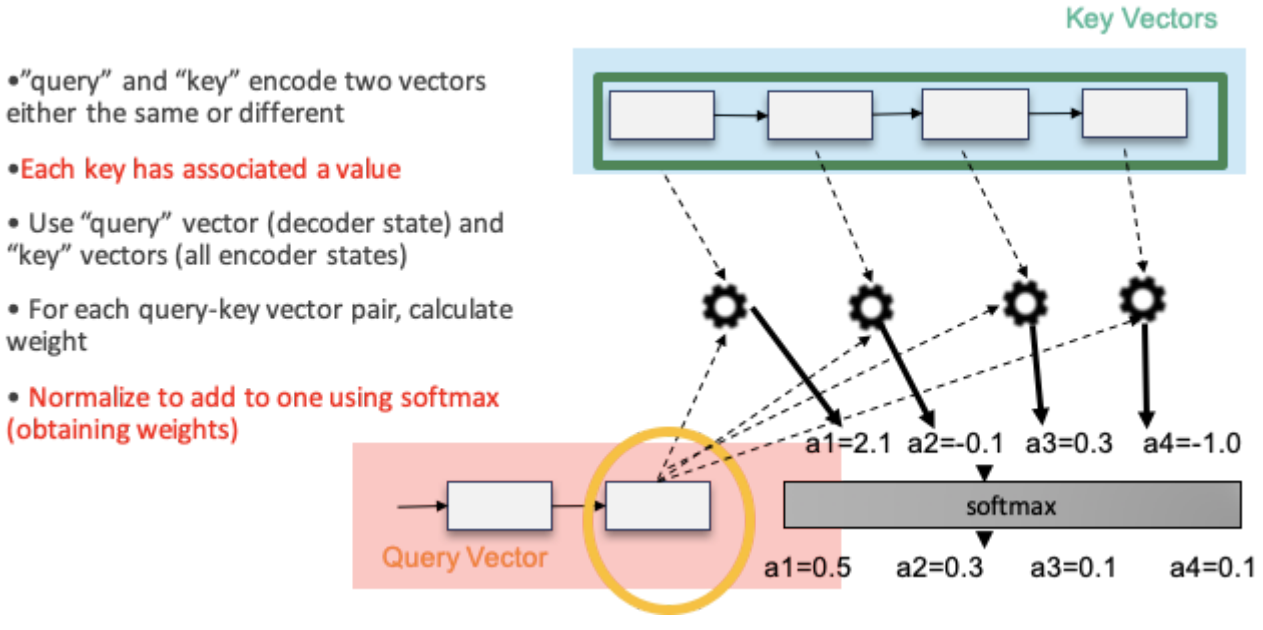
(C) The Transformer architecture uses **……..** in the encoder, **……..** attention and **…….** in the decoder
(D) Variations of attention-based mechanisms include: …… attention which is applied to speech recognition and coverage attention which deals with machine translation problems of **……** and **……**

Solution:
(A) True, it is the definition of attention
(B) False. Attention-based mechanisms have achieved great success in image and speech.
(C) self-attention; encoder-decoder; masked self-attention
(D) monotonic; over-translation and under-translation

D10L2 ATTENTION-BASED MODELS II by Marta R. Costa-jussà

Draw an scheme of how multiplicative attention is computed in an encoder-decoder architecture. Do not forget to mark the key and query vectors. Mention at least a couple of attention score functions.

Solution:



NAME OF THE STUDENT	

Multi-layer Perceptron	**a(q,k)=w_2^t tanh(W_1 [q;k])**	**Flexible, often very good with large data**	**Bahdanau et al., 2015**
Bilinear	**a(q,k)=q^T Wk**		**Luong et al 2015**
Dot Product	**a(q,k)=q^T k**	**No parameters! But requires sizes to be the same**	**Luong et al. 2015**
Scaled Dot Product	**a(q,k)=(q^T k)/√(|d_k |)**	**Scale by dimension of the key vector**	**Vaswani et al. 2017**

D11L1 REINFORCEMENT LEARNING by Xavier Giró

a) Given a state s, what function specifies which action a to take ?

b) What is the name of the function described in the following expression ?

$$\mathbb{E}_\pi[G_t|S_t = s]$$

c) What is the name of the function described in the following expression ?

$$\mathbb{E}_\pi[G_t|S_t = s, A_t = a]$$

d) What is the name of the function that describes how good or bad an action is with respect to the expectation of returns over all possible actions for a given state ? Provide its expression, too.

Solution:

d) The policy π.

e) State-value function.

$$V_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s]$$

f) The Q-value function or action-value function.

$$Q_\pi(s,a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a]$$

g) A-value (advantage) function.

$$A_\pi(s,a) = Q_\pi(s,a) - V_\pi(s)$$