Department of Electronic Systems

# Examination paper for TTT4185 Speech Technology

**Academic contact during examination::** Tor André Myrvoll
**Phone:** +47 95 14 80 14

**Examination date:** Thursday 14. December 2017

**Examination time (from - to):** 09.00 - 13.00

**Permitted examination support material:** **C** – Specified, written and handwritten examination support materials are permitted. A specified, simple calculator is permitted

**Other information:**

- The examination consists of 3 problems where
  - problem 1 concerns speech analysis
  - problem 2 concerns classification and speech recognition
  - problem 3 concerns deep learning
- All sub-problems counts the same
- All problems are to be answered
- Grades will be announced 3 weeks after the examination date.

**Language:** English

**Number of pages (front page excluded):** 9

**Number of pages enclosed:**

**Informasjon om trykking av eksamensoppgave**

**Originalen er :**

**1-sidig** □ **2-sidig** □

**sort/hvit** □ **farger** □

**skal ha flervalgsskjema** □

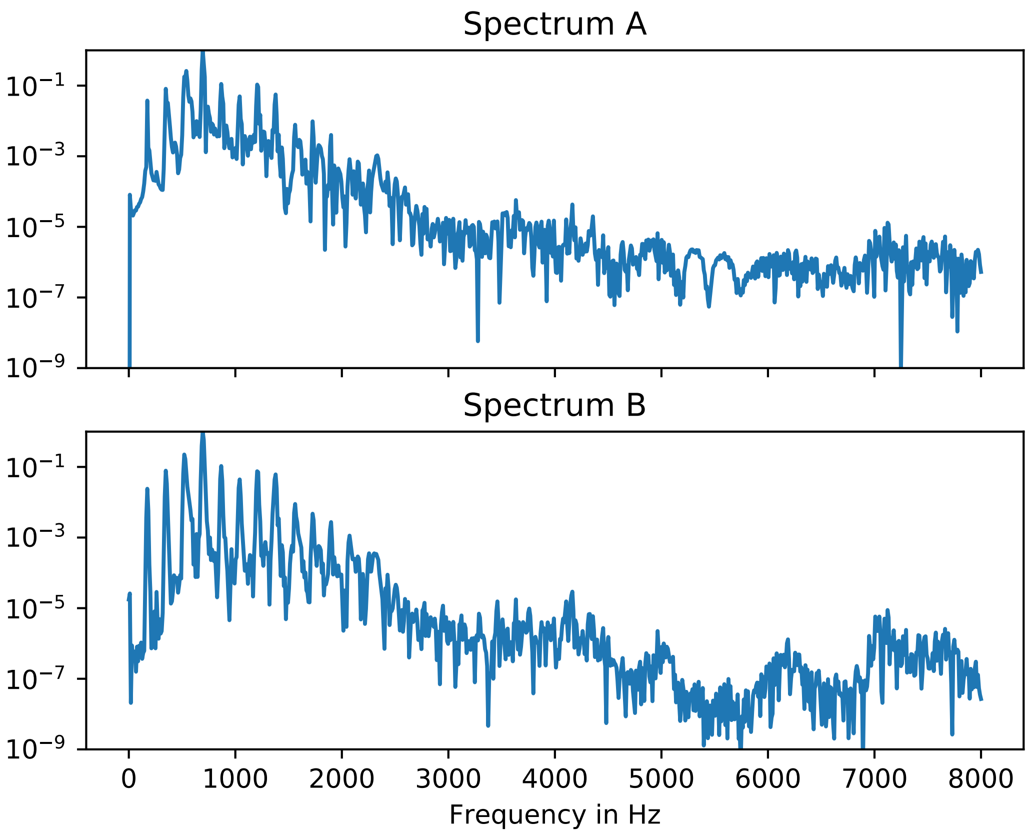
**Checked by:**

Date Signature

Note! Students will find the examination results in Studentweb. If you have questions about your results, you must contact your department. The Examination office will not be able to answer such inquiries.

## Problem 1

This set of problems addresses speech analysis.



**1a)** In the above figure there are two spectrums derived from the same speech signal. One is derived using a rectangular window, while the other is based on a Hamming window. In general, what is the main tradeoff one need to make when choosing a window for use with spectrum analysis? Use this insight to explain which signal is based on the Hamming window.

The tradeoff is between the width of the main lobe and the attenuation of the forst side-lobe of the frequency plane window. A wider lobe will decrease the resolution of the spectrum, while higher side-lobes will "smear" the spectrum, filling in valleys in the spectrum.

The Hamming window results in a wider main lobe and more attenuated side lobes when compared with the rectangular window. From this we conclude that Spectrum A is based on the rectangular window since the dips in the spectrum is partly filled in.

**1b)** Using either of the two spectrums, estimate the fundamental frequency of the speech signal. Is it likely to be a male or female voice?

We have 12 peaks over a range of 2 kHz, which means that $F_0 \approx 2000/12 = 167 Hz$. The pitch is low, making it likely to be a man.

**1c)** A way to visualize speech signals is by using *spectrograms*. Explain how a spectrogram is computed. What is the difference between a narrow- and a wide-band spectrogram?

The spectrogram is computed by computing the spectrum for small, overlapping frames of speech. The absolute value of the spectra are stacked contiguously to form an image where the $y$-axis is the frequency and the $x$-axis is time.

The choice of frame length determines whether the spectrogram is narrow- or broadband. Using longer frames gives a higher frequency resolution and could even resolve the $F_0$ harmonics. This is a narrowband spectrogram.

Using a shorter frame makes for a coarser spectrum, but the shorter frame facilitates higher resolution on the time axis. This is a broadband spectrogram.

**1d)** Another representation of speech signals is the *cepstrum*. Explain how the real cepstrum is computed, and how the cepstrum can be used to remove any effects of channels, e.g. the recording microphone, from the speech signal.

Let $|X(\omega)|$ be the absolute spectrum computed from the signal $x[n]$. Then the real cepstrum is defined as

$$\begin{aligned} c_x[k] &= \text{IDFT}\{\log|X(\omega)|\} \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \log|X(\omega)|e^{i\omega k}d\omega \end{aligned}$$

Now assume that $y[n] = x[n] \star h[n]$. Then

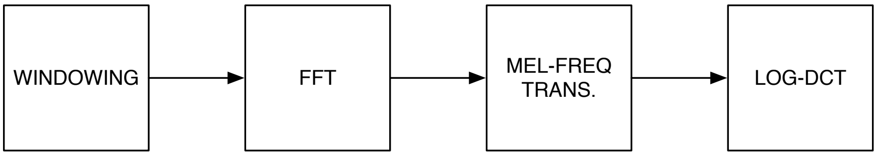
$$\begin{aligned} c_y(k) &= \text{IDFT}\left\{\log|Y(\omega)|\right\} \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \log|Y(\omega)|e^{i\omega k}d\omega \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \log|X(\omega)H(\omega)|e^{i\omega k}d\omega \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \left(\log|X(\omega)| + \log|H(\omega)|\right)e^{i\omega k}d\omega \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \log|X(\omega)|e^{i\omega k}d\omega + \frac{1}{2\pi}\int_{-\pi}^{\pi} \log|H(\omega)|e^{i\omega k}d\omega \\ &= c_x(k) + c_h(k) \end{aligned}$$

Since the channel is additive in the cepstral domain we can now estimate it by computing the average of $c_y(k)$ and subtracting it.

**1e)** What is the Mel frequency scale, and what perceptual observation is it based on?

The Mel frequency scale is a non-linear transformation of the frequency scale. It is based on the perceptual phenomenon that the human ear finds it increasingly harder to separate frequencies as we move up the spectrum. In other words – the ear becomes worse at spectrum analysis at high frequencies.

**1f)** The sampled speech waveform is not suited for classification tasks like automatic speech recognition. Instead, the speech signal is transformed into a sequence of *Mel frequency cepstral coefficients* (MFCC) feature vectors. Explain step by step how this process is done.



Prosesseringen skjer som følger:

- WINDOWING: Talesignalet deles opp i små, overlappende blokker som pålegges et vindu (typisk Hamming).
- FFT: Blokken transformeres til frekvensdomenet
- MEL-FREQ: Energien i overlappende delbånd av økende bredde beregnes. Delbåndenes posisjon og bredde er gitt av Mel-skalaen, en perseptuell frekvensskala.
- LOG-DCT: Energien i delbåndene representeres ved sin log-verdi og transformeres vha. en diskret cosinus transform. Dette er ment å dekorrelere egenskapsvektoren.

## Problem 2

This set of problems addresses fundamental classification problems and automatic speech recognition.

**2a)** You want to classify an observation $x$ into one of $M$ classes, $\{c_1, \ldots, c_M\}$. The Bayes classifier is defined as

$$\begin{aligned} m^\star &= \underset{m}{\text{argmax}}\, P(c_m|x) \\ &= \underset{m}{\text{argmax}}\, P(x|c_m)P(c_m). \end{aligned}$$

Explain the meaning of the different terms in the above definition. Also explain how we get from the first to the final expression. In what sense is this classifier optimal?

- $m^\star$ is the optimal class
- $P(c_m|x)$ is the *a posteriori probability* of class $c_m$ given a known observation $x$.
- $P(x|c_m)$ is the probability of observing $x$ given that class $c_m$ is the true class
- $P(c_m)$ is the *a priori probability* of class $c_m$. This is the probability one assign to $c_m$ being true given no observations.

The final expression is found using Bayes rule

$$\begin{aligned} m^\star &= \underset{m}{\text{argmax}}\, P(c_m|x) \\ &= \underset{m}{\text{argmax}}\, \frac{P(x|c_m)P(c_m)}{P(x)} \\ &= \underset{m}{\text{argmax}}\, P(x|c_m)P(c_m). \end{aligned}$$

The final equality stems from the fact that $P(x)$ does not depend on $m$ and can be ignored.

The Bayes classifier is optimal in the sense that it minimizes the mis-classification rate (error rate).

**2b)** An automatic speech recognizer is in its fundamental form based on the Bayes classifier, with the speech signal modeled by an *acoustic model* and the word sequence by a *language model*. Write down the Bayes classifier in terms of these two models.

Let $X$ be the acoustic signal and $W$ the underlying string of words. Then the Bayes classifier is

$$\begin{aligned} W^\star &= \underset{W}{\text{argmax}}\, P(W|x) \\ &= \underset{W}{\text{argmax}}\, P(X|W)P(W) \end{aligned}$$

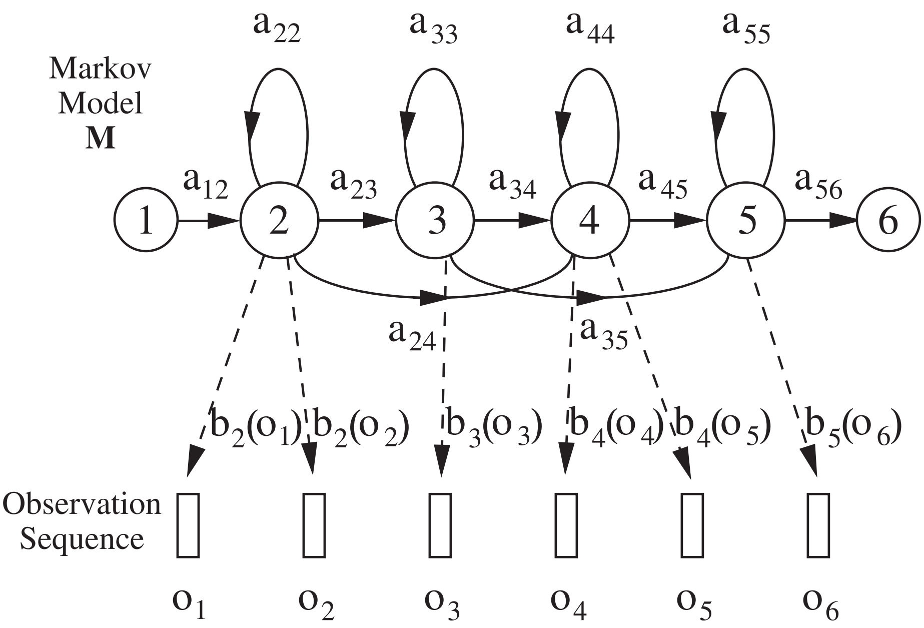
Here $P(X|W)$ is the acoustic model, and the $P(W)$ is the language model.

**2c)** The acoustic model is usually based on the *hidden Markov model*. Describe the hidden Markov model, including its parameters, and how you would use an HMM as a model for a single phoneme.

An HMM is based on a sequency of discrete states that is observed indirectly through a random observation whose distribution is depending on the current state.

If $\{1, 2, \ldots, S\}$ is a set of states, the sequence of states is a random sequencw where probability of the next state depends on the current state only. This is a Markov chain.

The observation is a random variable as well, and is only depending on the current state. An example of a four state HMM is given below:



For a single foneme we should use a three state model where the states correspond to the beginning, middle and end of the phoneme. The state transition probabilities should be chosen so that the the probability of jumping ”backwards”, eg. from the end to the start again is zero.

**2d)** The word string $W$ is typically modeled by an $N$-*gram*. Describe the $N$-gram model. What is the main challenge when estimating $N$-gram parameters for moderate to large $N$? Explain how discounting, amortization and interpolation addresses this problem.

The $n$-gram model models the probability of a given word in an utterance as a discrete probability given the $n - 1$ previous words in the utterance. Example of a 3-gram:

$$\begin{aligned}&P(\text{I would like to listen to some Jazz})\\ =&P(\text{I})P(\text{would}|\text{I})P(\text{like}|\text{would, I})P(\text{to}|\text{like, would})P(\text{listen}|\text{to, like})P(\text{to}|\text{listen, to})\\ &\times P(\text{some}|\text{to, listen})P(\text{Jazz}|\text{some, to})\end{aligned}$$

*Discounting*, *backoff* and *interpolation* are used to address the problem of estimating eg. 64 trillion parameters for a trigram (there is just not enough data in the world. Also, you wouldn’t want to store all those parameters). In practice most trigrams would have zero examples in a limited training set, giving them a probability of zero, which would be a problem if that trigram was encountered during use.

- Discounting: Keep a small probability mass in reserve and distribute it evenly across all unseen trigrams.
- Backoff: If the trigram isn't found – use the bigram. If the bigram is not found – use the unigram.
- Interpolation: $P_I(w_3|w_2, w_1) = \alpha P(w_3|w_2, w_1) + \beta P(w_3|w_2) + (1 - \alpha - \beta)P(w_3)$ The weights $alpha, \beta$ are usually trained to maximize perplexity.

## Problem 3

This set of problems addresses the theory of deep learning for neural networks, and its application to speech recognition.

**3a)** Consider a Multilayer Perceptron (MLP) having 5 layers – one input layer, three hidden layers and one output layer. The number of nodes in each layer is three, four, four, four and two (3,4,4,4,2) respectively. How many parameters need to be estimated during training?

We need to estimate

$$3 \times 4 + 4 \times 4 + 4 \times 4 + 4 \times 2 = 52$$

connection parameters and

$$4 + 4 + 4 + 2 = 14$$

bias parameters. All in all 66 parameters.

**3b)** The network is to be used as a binary classifier. What is a reasonable choice for the function to use in the output nodes in this case? What would you use if the problem was of a regression type?

We want a function that guarantees that the outputs are all positive and sums to one (ie. a probability distribution). The softmax can be used

$$z_i = \frac{e^{v_i}}{\sum_j e^{v_j}},$$

where $\{e_j\}$ is the softmax input from the network and $z_i$ is the $i$th output.

For regression one should use a linear layer,

$$z_i = v_i$$

**3c)** You are given a set of training data $\{x_n, y_n\}$, $n = 1 \ldots N$, and a loss function, $l(\cdot, \cdot)$, that measures the loss of the classification made by the network with input $x$, given the true value $y$. Write down the empirical loss for the training set.

For every observation $x_n$ let $\hat{y}_n = f(x_n)$ be the output of the neural network. Then the empirical loss is

$$L = \sum_n l(\hat{y}_n, y_n) \tag{1}$$

**3d)** Training a neural network is usually done using gradient descent, that is

$$\Theta_{t+1} = \Theta_t - \varepsilon_t \Delta\Theta$$

where $\Theta_t$ is the model parameters at time $t$ and $\Delta\Theta$ is the derivative of the empirical loss with respect to the model parameters. Give a high level explanation on how the back-propagation algorithm is used to this end.

Let $f(\cdot; \Theta)$ be the mapping that the neural network represents. The back propagation algorithm first performs a *forward pass*, where $\hat{y}_n = f(x_n; \Theta_t)$ is computed for all $n$, keeping all intermediate results from inside the network.

The backpropagation algorithm then starts by computing the gradient with respect to the final layer in the network. The gradient wrt. the next-to-final-layer can be written in terms of the gradient from the final layer and the stored results from the forward pass.

In the same way, the gradient wrt. every layer can be computed in terms of the previous layer, hence the term back-propagation.

**3e)** When training a neural network using gradient descent, it is done using approaches referred to as batch, mini-batch or stochastic gradient descent. Explain what we mean by these methods, and describe their strengths and weaknesses.

- Batch: Gradienten beregnes ut fra alle eksemplene i treningssettet.
  - Fordel: Dette er den korrekte gradienten ut fra ønsket om å optimalisere kostfunksjonen på treningssettet.
  - Ulempe: Tar lang tid å beregne. Lite effektivt.
- SGD: Gradienten beregnes ut fra et eneste eksempel fra treningssettet
  - Fordel: Meget enkel og effektiv å beregne. Konvergerer raskere enn batch på store datasett
  - Ulempe: Vanskelig å parallelisere
- Minibatch: Beregn gradient vha. et lite subset av treningsdataene.
  - Fordel: Som SGD – mer effektiv enn batch for store datasett. Kan enkelt paralleliseres.
  - Ulempe: Ingen signifikante

**3f)** The use of Deep learning and neural networks has brought about a significant improvement in automatic speech recognition systems. Give an overview of the CD-DNN-HMM (context dependent, deep neural network, hidden Markov model) as compared to the classical GMM-HMM (Gaussian mixture model, hidden Markov model).

Et *senon* er en tilstand i et trifon, og er ofte er delt mellom flere trifoner. Siden antall unike tilstander som trengs for å modellere en trifon-modell er svært høyt, kan man la flere trifoner dele en eller flere tilstander.

En CD-DNN-HMM er basert på et dypt nevralt nettverk som klassifiserer tale som senoner. Med andre ord – gitt en observasjonsvektor $o_t$, gir utgangen av et DNN oss sannsynlighetene, $P(s_t|o_t))$, for alle senonene i modellen.

En klassisk GMM-HMM baserer seg på observasjonssannsynlighetene $P(o_t|s_t)$, men disse kan skrives som

$$P(o_t|s_t) = \frac{P(s_t|o_t)P(o_t)}{P(s_t)} \tag{2}$$

Siden $P(o_t)$ er konstant for alle $s_t$ kan den ignoreres, og vi kan bruke

$$\bar{P}(o_t|s_t) = \frac{P(s_t|o_t)}{P(s_t)} \tag{3}$$

direkte i vårt standard GMM-HMM rammeverk.

**3g)** The current research front in automatic speech recognition tries to replace the hidden Markov model (HMM) completely, instead using a combination of convolutional neural networks (CNN) and recurrent neural networks (RNN). Describe the CNN and its special layers, and explain the general motivation for using CNNs. Also describe the RNN and explain how it can be used to process sequences of arbitrary length.

The main motivation behind the CNN is the reduction in complexity. For very large input observations, eg. images, a fully connected network is infeasible. Instead, each node is only connect to a small subset of nodes in the next layer. Also, the weights are reused, resulting in a convolution-like operation that reduces complexity significantly while keeping performance.

In addition to the convolution layer a CNN uses poolingl layers and processing layers. The pooling layer takes a number of inputs and only outputs eg. the maximum or some other summary statistic. This reduces the size of the network as well as introducing some translation and rotation invariance. Finally, the processing layer is a layer with standard non-linearities like sigmoids and ReLUs.

The RNN (Recurrent Neural Network) has a feedback from and to its hidden nodes. This enables the hidden nodes to have some memory about previous observation and function as a state. This enables the network to process sequences of observations, updating its state continuosly.