Exam

- **Time**: 90 mins. Perhaps some multiple choice questions to warm up, but mostly free-text.
- **Content**: Lectures 1-10 excluding TICA, TCCA, Boltzmann Machines
- **Understand** of Concepts / Theory:
  - Estimator theory basics (Variance+Bias, Generalization, Over/Underfitting)
  - Universal Representation Theorem
  - Local minimization, Backpropagation
  - Gradient annihilation in deep learning and methods to fix it.
- **Understand** of Algorithmic aspects:
  - Training and Validation set.
  - One-hot encoding.
  - Common losses (mean square error, cross-entropy).
  - Early stopping, Data augmentation, Parameter sharing, Dropout.
  - Common choices of nonlinearities (Identity, Binary, Logistic, ReLU).
  - Minimization methods: Gradient descent, Stochastic Gradient descent, Basic idea of adaptive methods (Adam, RMSprop).
  - Convolution, transposed Convolution, Pooling, Padding.
  - Probabilistic interpretation of certain units/layers and losses (Sigmoid, SoftMax, Cross-Entropy).

Exam

- **Time**: 90 mins. Perhaps some multiple choice questions to warm up, but mostly free-text.
- **Content**: Lectures 1-10 excluding TICA, TCCA, Boltzmann Machines
- **Understand** of Concepts / Theory:
  - Estimator theory basics (Variance+Bias, Generalization, Over/Underfitting)
  - Universal Representation Theorem
  - Local minimization, Backpropagation
  - Gradient annihilation in deep learning and methods to fix it.
- **Understand** of Algorithmic aspects:
  - Training and Validation set.
  - One-hot encoding.
  - Common losses (mean square error, cross-entropy).
  - Early stopping, Data augmentation, Parameter sharing, Dropout.
  - Common choices of nonlinearities (Identity, Binary, Logistic, ReLU).
  - Minimization methods: Gradient descent, Stochastic Gradient descent, Basic idea of adaptive methods (Adam, RMSprop).
  - Convolution, transposed Convolution, Pooling, Padding.
  - Probabilistic interpretation of certain units/layers and losses (Sigmoid, SoftMax, Cross-Entropy).

Exam

- **Time**: 90 mins. Perhaps some multiple choice questions to warm up, but mostly free-text.
- **Content**: Lectures 1-10 excluding TICA, TCCA, Boltzmann Machines
- **Understand** of Concepts / Theory:
  - Estimator theory basics (Variance+Bias, Generalization, Over/Underfitting)
  - Universal Representation Theorem
  - Local minimization, Backpropagation
  - Gradient annihilation in deep learning and methods to fix it.
- **Understand** of Algorithmic aspects:
  - Training and Validation set.
  - One-hot encoding.
  - Common losses (mean square error, cross-entropy).
  - Early stopping, Data augmentation, Parameter sharing, Dropout.
  - Common choices of nonlinearities (Identity, Binary, Logistic, ReLU).
  - Minimization methods: Gradient descent, Stochastic Gradient descent, Basic idea of adaptive methods (Adam, RMSprop).
  - Convolution, transposed Convolution, Pooling, Padding.
  - Probabilistic interpretation of certain units/layers and losses (Sigmoid, SoftMax, Cross-Entropy).

Exam

- **Time**: 90 mins. Perhaps some multiple choice questions to warm up, but mostly free-text.
- **Content**: Lectures 1-10 excluding TICA, TCCA, Boltzmann Machines
- **Understand** of Concepts / Theory:
  - Estimator theory basics (Variance+Bias, Generalization, Over/Underfitting)
  - Universal Representation Theorem
  - Local minimization, Backpropagation
  - Gradient annihilation in deep learning and methods to fix it.
- **Understand** of Algorithmic aspects:
  - Training and Validation set.
  - One-hot encoding.
  - Common losses (mean square error, cross-entropy).
  - Early stopping, Data augmentation, Parameter sharing, Dropout.
  - Common choices of nonlinearities (Identity, Binary, Logistic, ReLU).
  - Minimization methods: Gradient descent, Stochastic Gradient descent, Basic idea of adaptive methods (Adam, RMSprop).
  - Convolution, transposed Convolution, Pooling, Padding.
  - Probabilistic interpretation of certain units/layers and losses (Sigmoid, SoftMax, Cross-Entropy).

Exam

- **Understand** Learning frameworks:
  - Regression and logistic Regression.
  - Perceptron, Multi-layer Perceptron.
  - ConvNet.
  - Autoencoder and time-Autoencoder, PCA.
  - RNN, Universal RNN, Teacher forcing, Multilayer RNNs, LSTM (basic ideas).
- **Be able to**:
  - Multivariate calculus (derivatives and integrals), and linear algebra (matrix-vector operations)
  - Reading and writing basic Python code
  - Being able to design learning structures with the learnt elements given a ML problem.
- We will **not** ask for:
  - Knowing non-fundamental technical details or complicated expression by heart.
  - Advanced decompositions (SVD).
  - Knowing details such as integrals for certain functions or all the gates in a LSTM by heart. We will give you these details if they are relevant to solve an exam problem.

Exam

- **Understand** Learning frameworks:
  - Regression and logistic Regression.
  - Perceptron, Multi-layer Perceptron.
  - ConvNet.
  - Autoencoder and time-Autoencoder, PCA.
  - RNN, Universal RNN, Teacher forcing, Multilayer RNNs, LSTM (basic ideas).
- **Be able to**:
  - Multivariate calculus (derivatives and integrals), and linear algebra (matrix-vector operations)
  - Reading and writing basic Python code
  - Being able to design learning structures with the learnt elements given a ML problem.
- We will **not** ask for:
  - Knowing non-fundamental technical details or complicated expression by heart.
  - Advanced decompositions (SVD).
  - Knowing details such as integrals for certain functions or all the gates in a LSTM by heart. We will give you these details if they are relevant to solve an exam problem.

Exam

- **Understand** Learning frameworks:
  - Regression and logistic Regression.
  - Perceptron, Multi-layer Perceptron.
  - ConvNet.
  - Autoencoder and time-Autoencoder, PCA.
  - RNN, Universal RNN, Teacher forcing, Multilayer RNNs, LSTM (basic ideas).
- **Be able to**:
  - Multivariate calculus (derivatives and integrals), and linear algebra (matrix-vector operations)
  - Reading and writing basic Python code
  - Being able to design learning structures with the learnt elements given a ML problem.
- We will **not** ask for:
  - Knowing non-fundamental technical details or complicated expression by heart.
  - Advanced decompositions (SVD).
  - Knowing details such as integrals for certain functions or all the gates in a LSTM by heart. We will give you these details if they are relevant to solve an exam problem.