ELEC-C5220
Lecture 6:
Speech Enhancement with Denoising Autoencoders

Machine learning in information technology

Aalto University
School of Electrical
Engineering

Lauri Juvela

15.2.2024

Lecture 06 content

- Speech denoising and enhancement
- Autoencoders
- Denoising autoencoders
- Data augmentation
- U-Net architecture
- Course project

What can go wrong in a video call?

- Let’s share worst experiences

What can go wrong in a video call?
Fix it with enhancement

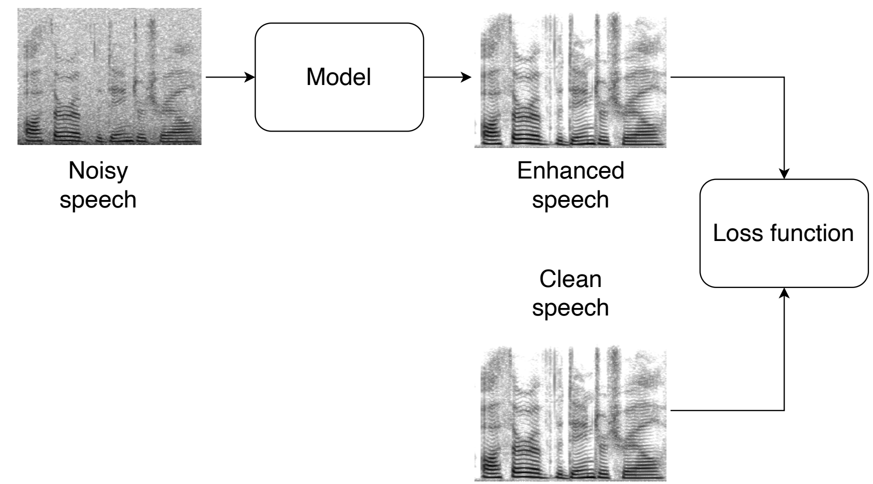
- Background noise
- Too much reverberation
- Distorting and clipping microphones
- Audio dropout
- Frame rate drops
- Pixelated and blurry video
- Poor lighting conditions

Denoising vs. enhancement

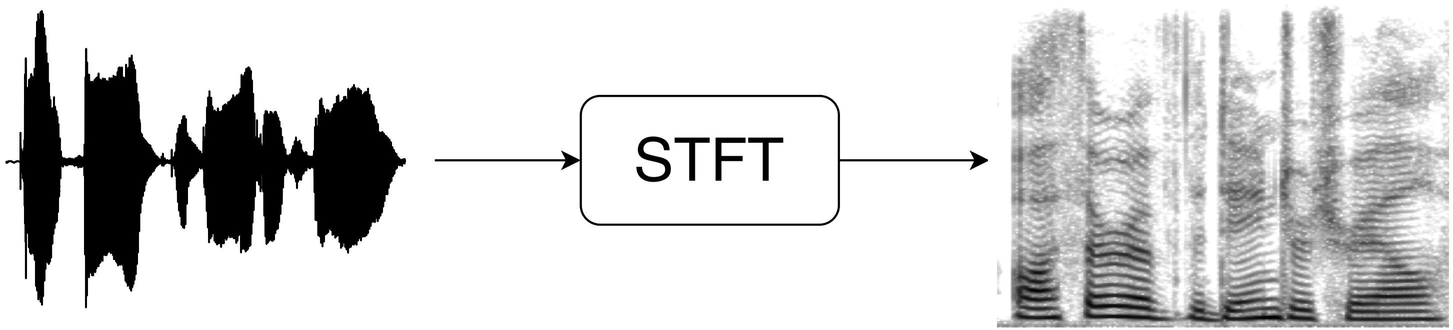
- Denoising is noise removal
- Enhancement includes noise removal

How to implement denoising?

- **Classic approach: estimate what is signal and what is noise, filter out the noise**
- **Pure Deep Learning approach: use a Neural Network model, noisy signal goes in, clean signal comes out**
- **Hybrid approach: use a NN model to predict a filter mask**



Spectrograms



Waveform

(Batch, Channels, Time)

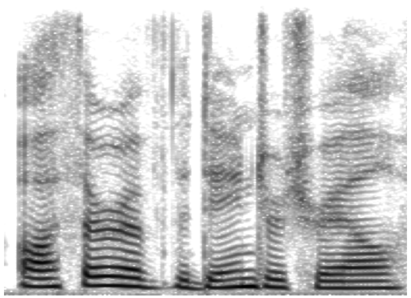
(1, 1, 24800)

Spectrogram

(Batch, Channels, Frequency, Time)

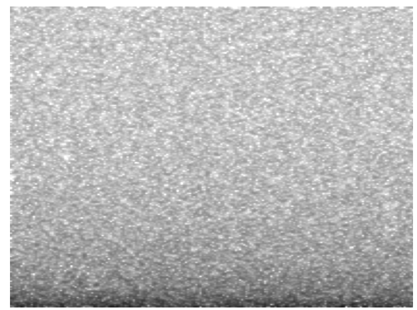
(1, 1, 257, 194)

Noise is commonly modeled as additive



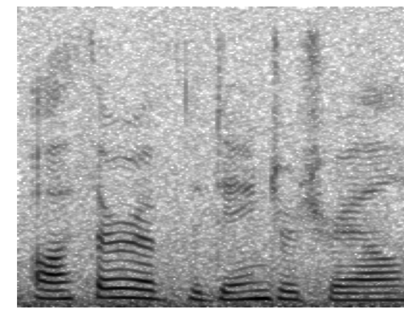
Clean speech

+



Noise

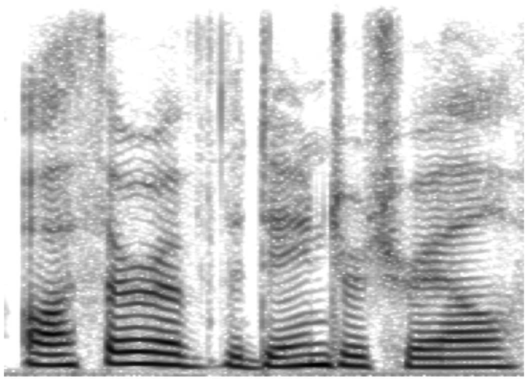
=



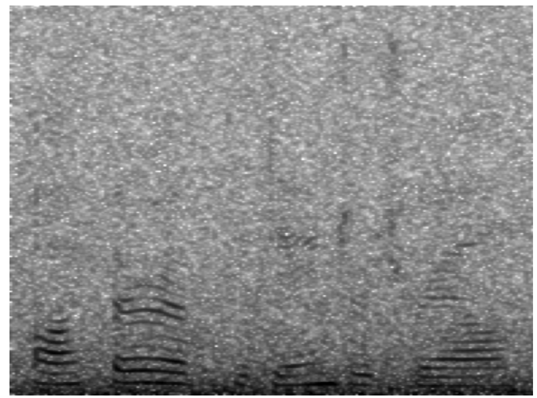
Noisy speech

Different noise types

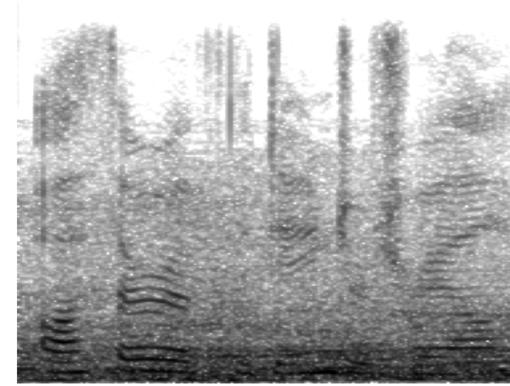
Clean speech



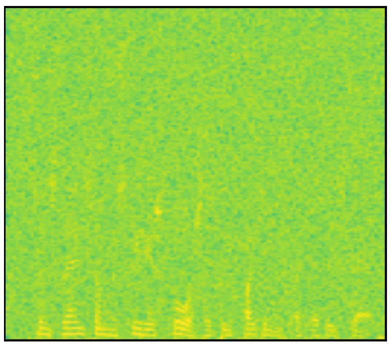
Speech in pink noise



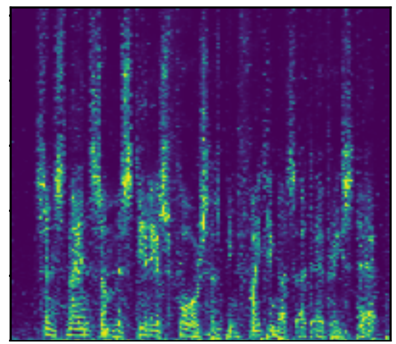
Speech in babble noise



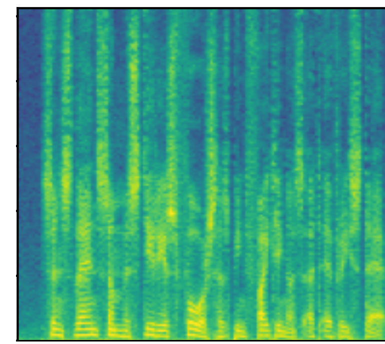
Multiplication with a frequency mask is filtering



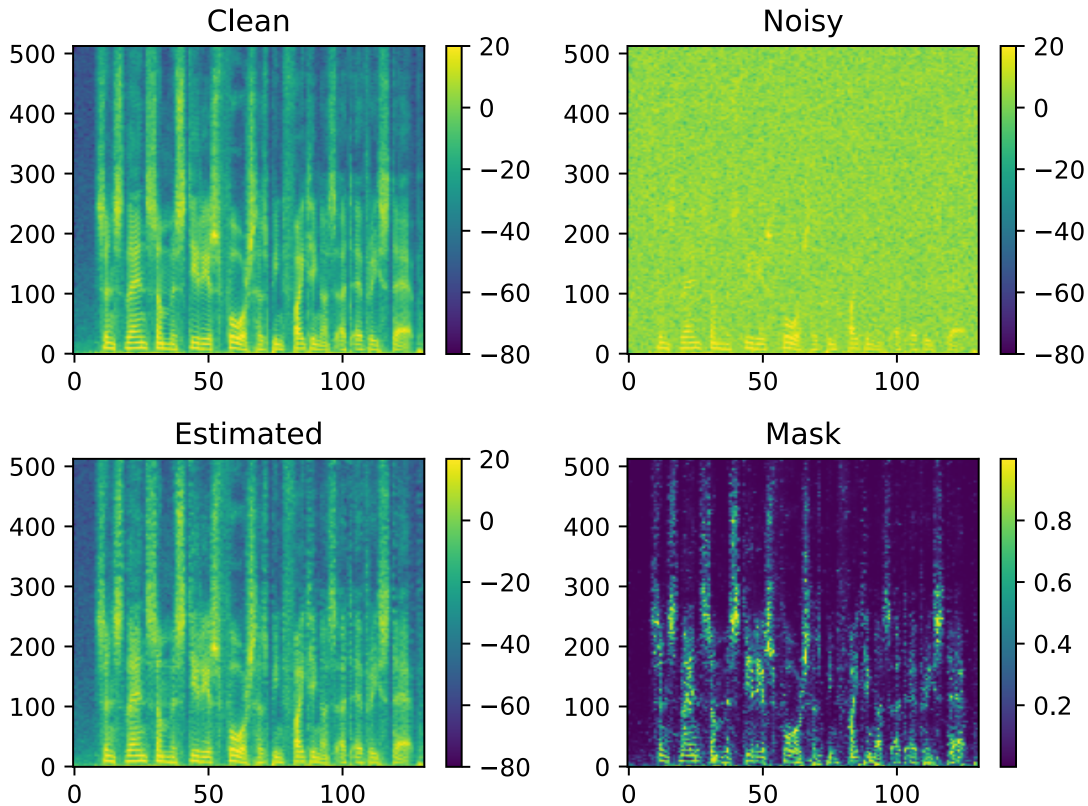
$\odot$



$=$



Mask estimation in Exercise 05



Spectral subraction

- **Works well for stationary noise**
- **Requires an estimate for noise power spectrum**

$$|\hat{S}(\omega)|^2 = |Y(\omega)|^2 - |N(\omega)|^2$$

Clean speech estimate	Noisy speech	Noise

Wiener filtering

- **Spectrogram masking**
- **Mask is close to zero when noise energy is large relative to signal**
- **Mask is close to on when signal energy is large relative to noise**
- **Compare with learning masks in Exercise 05**

$$H_{\text{Wiener}}(\omega) = \frac{|S(\omega)|^2}{|S(\omega)|^2 + |N(\omega)|^2}$$

$$|\hat{S}(\omega)|^2 = H_{\text{Wiener}}(\omega)|Y(\omega)|^2$$

Additivity and Fourier transforms

- Fourier transforms are linear – additivity is preserved
- Problem: transformed variables are complex valued!

$$x(t) + y(t) = z(t)$$

$$X(z) + Y(z) = Z(z)$$

Additivity and Fourier transforms

- **Power spectrum of a sum includes a cross- correlation term**

$$|X(z)|^2 + |Y(z)|^2 + 2X^*(z)Y(z) = |Z(z)|^2$$

- **Power spectra are additive for uncorrelated signals**

$$|X(z)|^2 + |Y(z)|^2 \hat{=} |Z(z)|^2$$

- **Spectrum magnitudes are not additive**
  - Often models that assume this work just fine, though

$$|X(z)| + |Y(z)| \neq |Z(z)|$$

Modeling phase is difficult

- **Use original (noisy) phase, modify the magnitude**
- **We did this in Exercise 05**

$$\hat{S}(\omega) = |\hat{S}(\omega)| e^{i\angle Y(\omega)} = \frac{|\hat{S}(\omega)|}{|Y(\omega)|} Y(\omega)$$

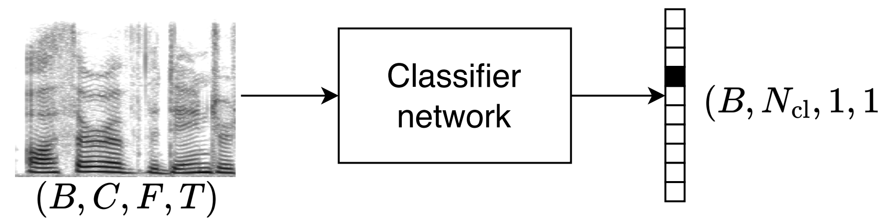
Modified complex spectrum

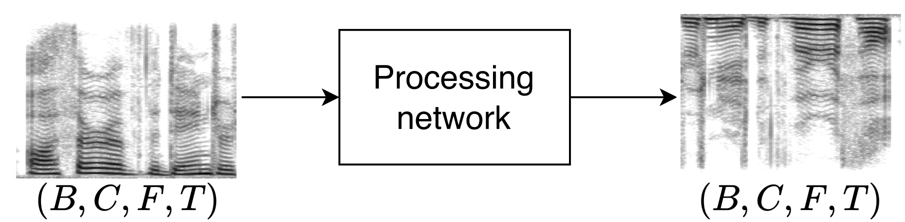
Modified magnitude spectrum

Original phase spectrum

What kind of models do we need?

- **Previously, we have worked with classifiers for dimensionality reduction**
- **Now we need to output the same shape as the input**



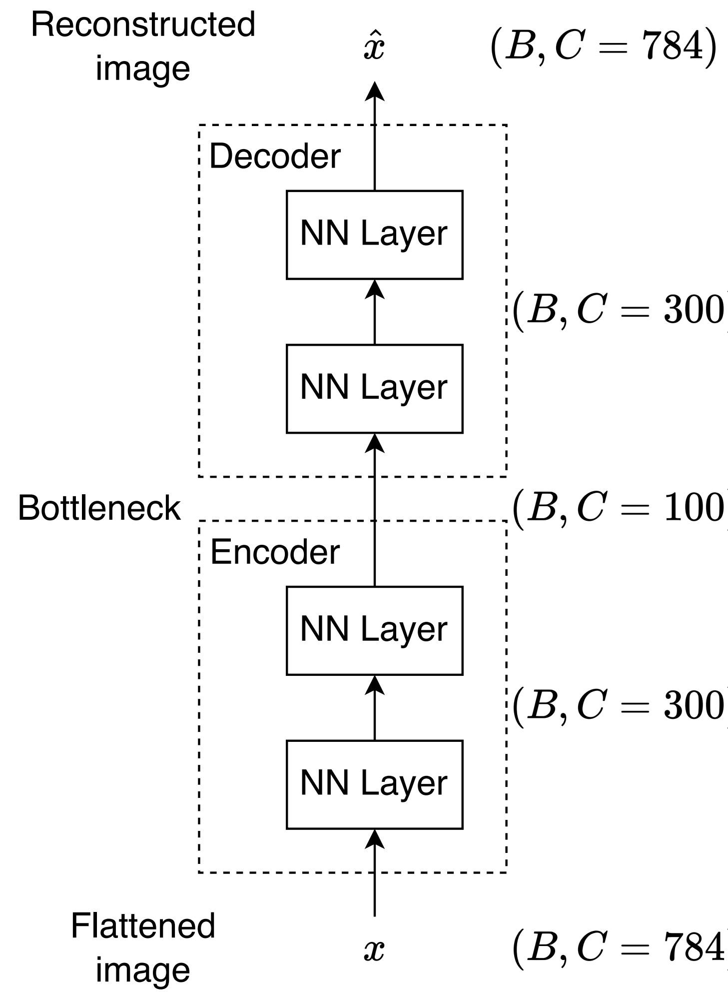


Encoder-Decoder models

- Learn to compress high dimensional data to a low-dimensional space and decompress it back to original data domain
- Similar idea to image, audio, and speech coding (JPEG, MP3, CELP)
- Information bottleneck principle: model uses its capacity to learn relevant features for the task and reject irrelevant features like noise

DNN Autoencoder

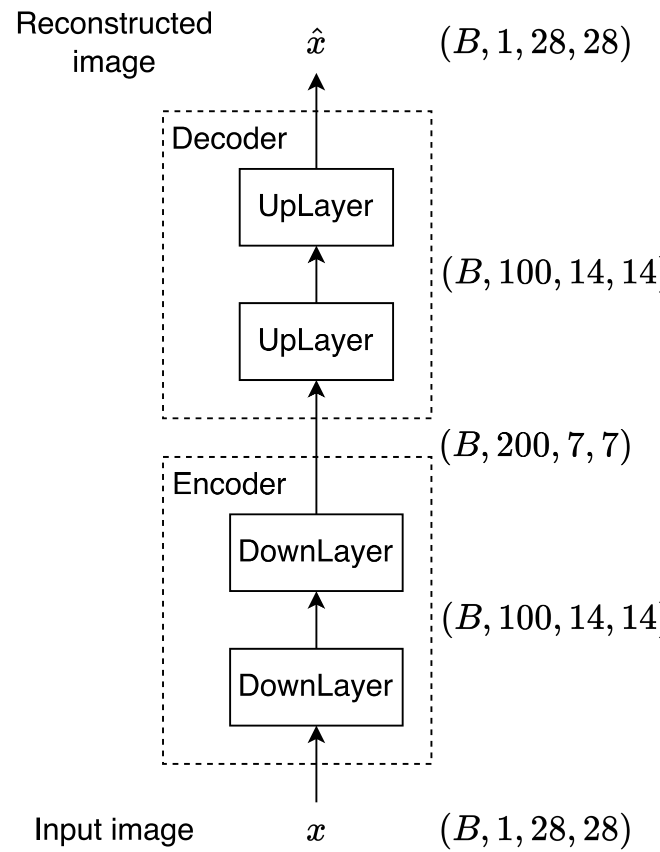
- **Data compression with neural networks**
- **Encoder reduces data dimensionality**
- **Decoder maps back to orignal data dimension**
- **Train to match reconstructed output with input**



Original and Autoencoded digits

CNN Autoencoder

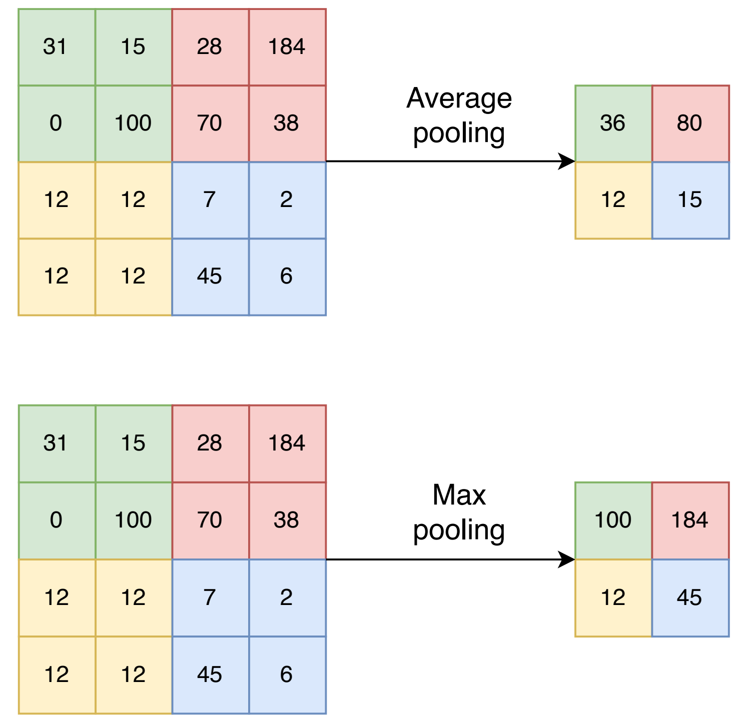
- **Similar idea**
- **Encoder applies spatial dimensionality reduction by downsampling**
- **Decoder reconstructs the spatial dimensions by upsampling**



$(\mathrm{Batch}, \mathrm{Channels}, \mathrm{Height}, \mathrm{Width})$

Downsampling

- **We have already used pooling layers for downsampling**
- **Convolution and pooling can be combined with *strided convolution***
- **Strided convolution is weighted average pooling with learnable weigthts**

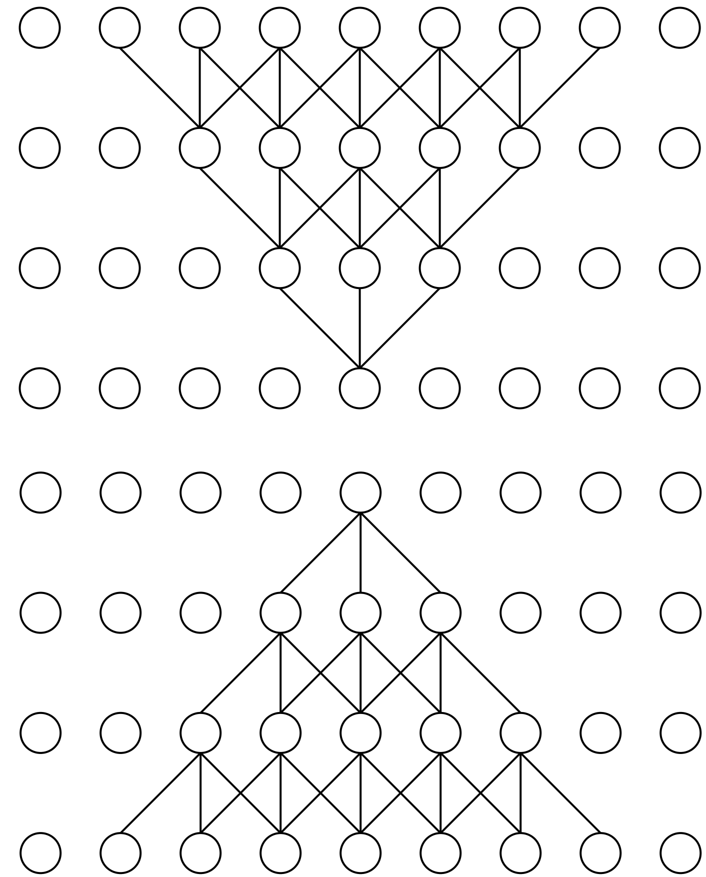


Upsampling

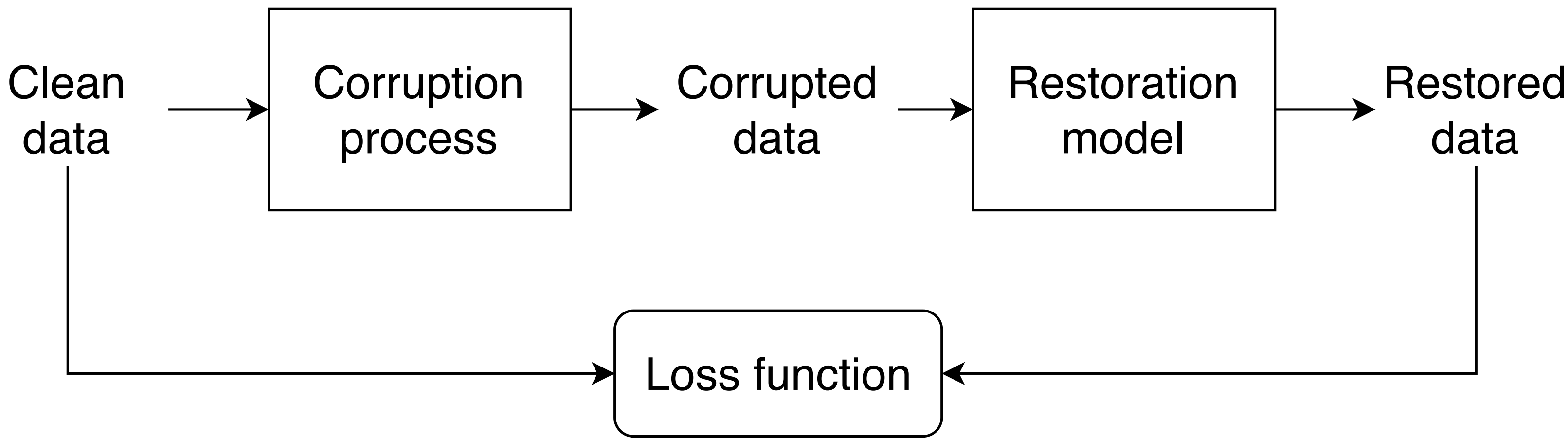
- Repeat values (nearest neighbor interpolation)
- Interpolation (linear, polynomial, sinc, etc.)
- Transposed Convolution

Transposed convolution

- **Learnable upsampling**
- **Connectivity pattern is mirrored from regular convolution**
- **Convolution is a linear operator and can be represented by as a matrix**
- **Transposed convolution corresponds to the transpose of the convolution matrix**

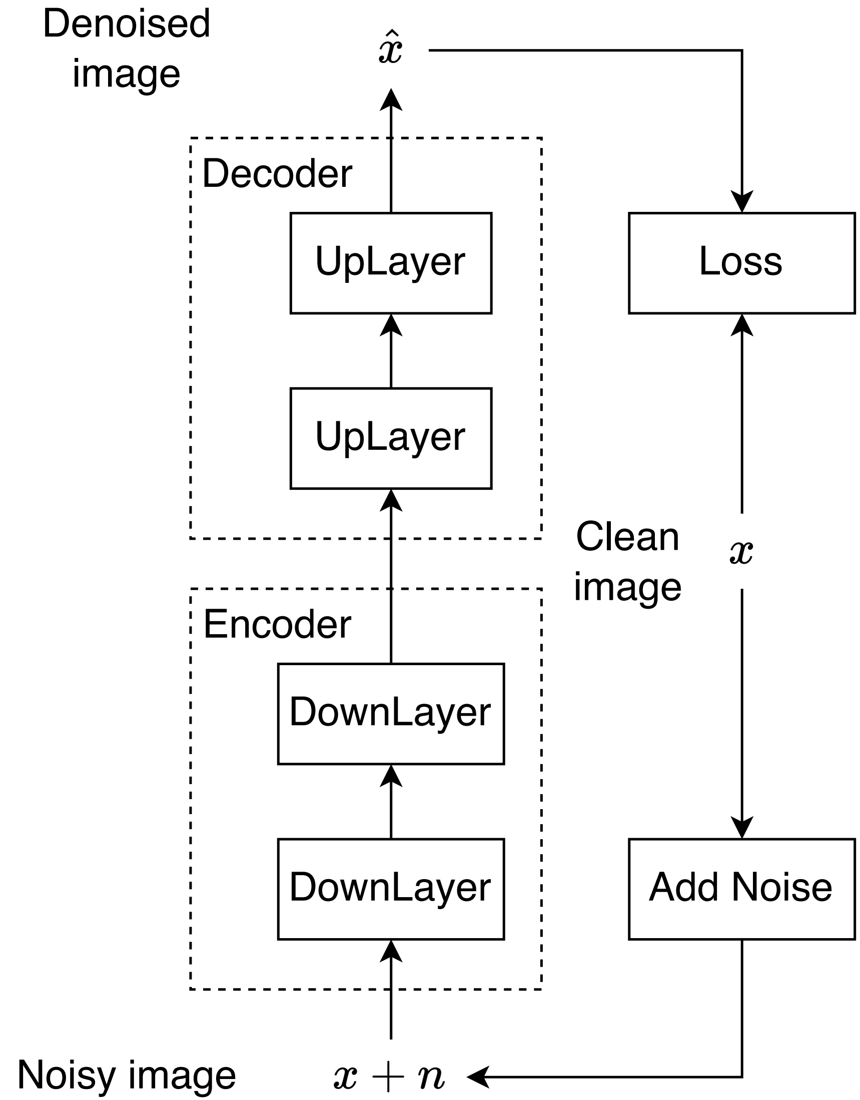


General enhancement workflow

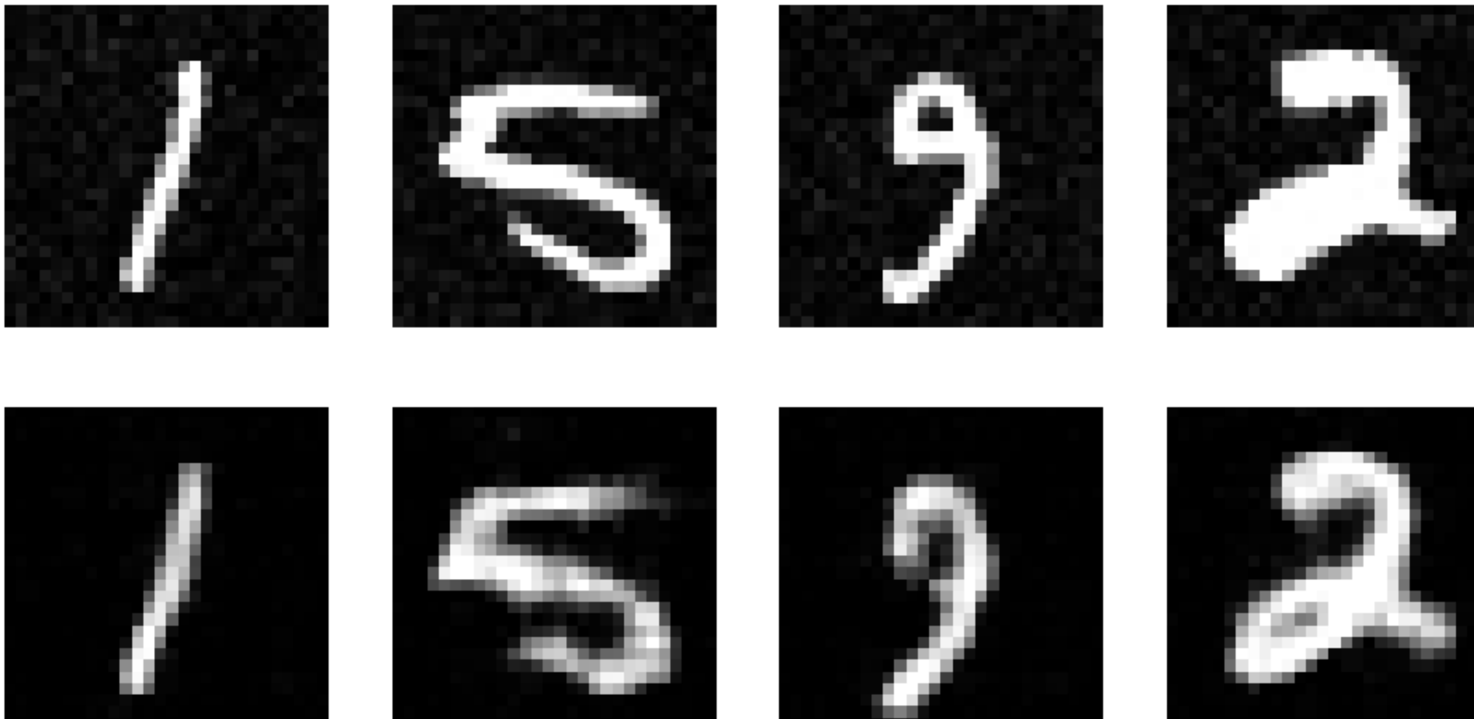


Denoising Autoencoder

- Add noise to clean data
- Encoder compresses
- Decoder decompresses
- Teach the system to remove noise
- Least-squares in pixel or waveform domain is common but not the best (see Exercise 05)

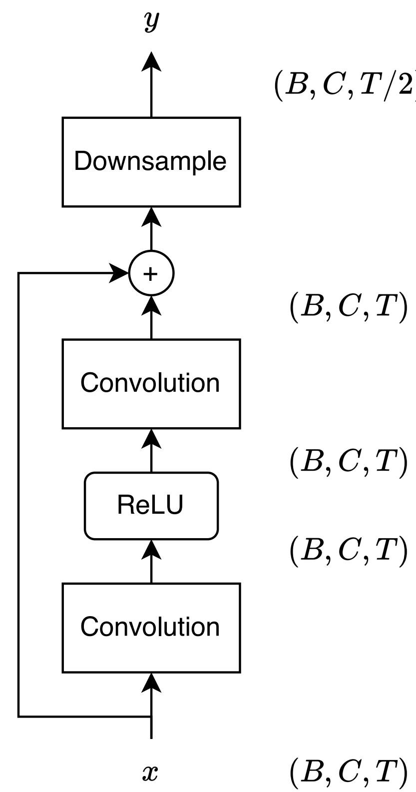


Noisy inputs and denoising autoencoder outputs



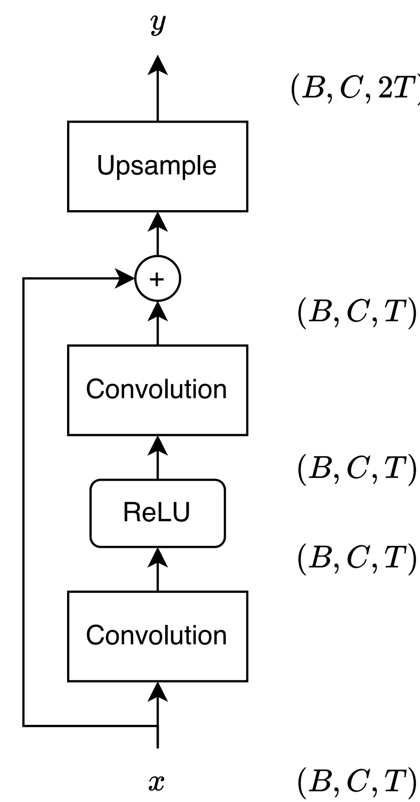
Convolution Block with Downsampling

- **Familiar from handwritten digit classification and spoken digit classification (Lecture and Exercise 02)**
- **Use this kind of blocks to build an encoder model**



Convolution Block with Upsampling

- **Similar to encoder building blocks, just replace downsampling with upsampling and mirror the structure**
- **Use this kind of blocks to build a decoder model**

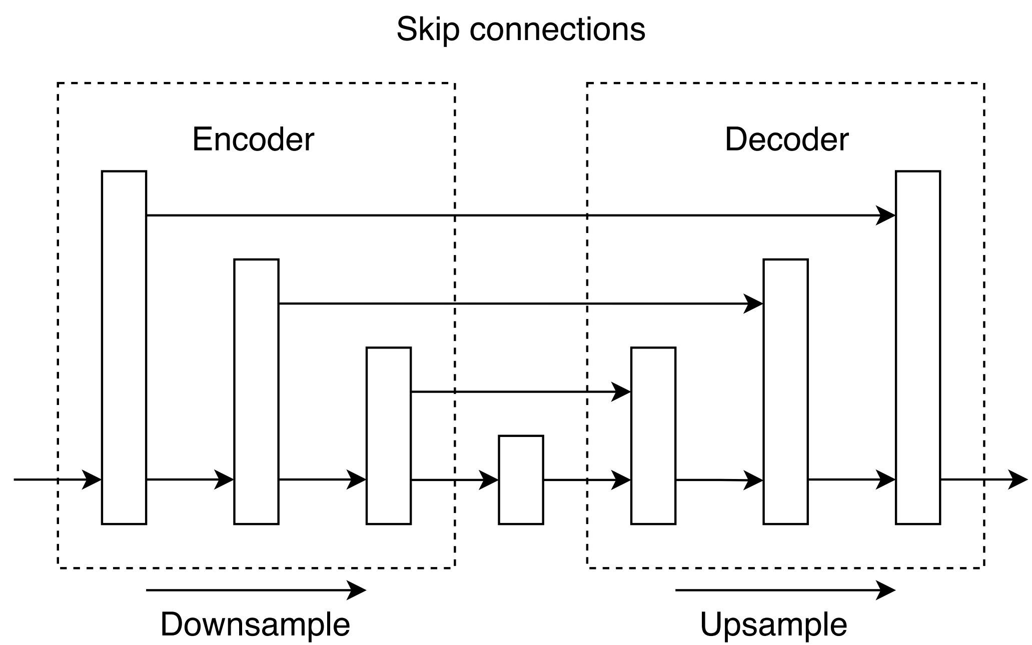


Data augmentation

- **Data augmentation refers to applying transformations to input data to artificially increase data diversity**
  - Noise, cropping, rotation, distortion etc.
  - Does not need to be differentiable (usually)
- **In classification, corrupting the input can improve robustness and generalization**
- **In enhancement and denoising, data augmentation is used to construct the training data**

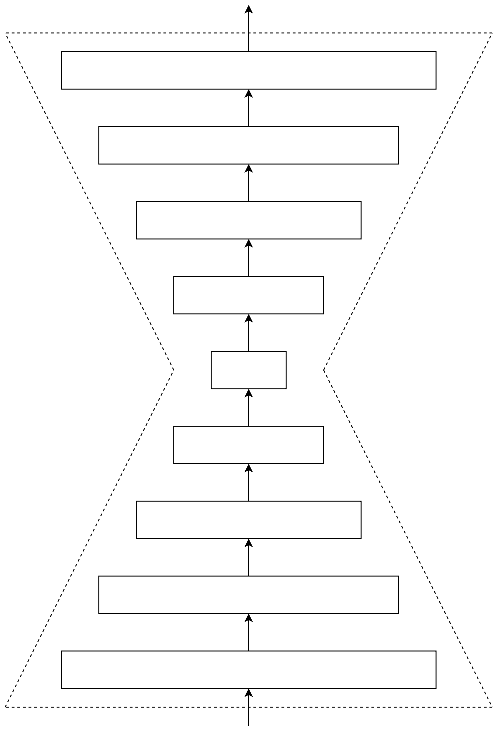
U-Net

- **U-Net architectures add skip connections between matched resolutions in Encoder and Decoder**
- **Analogous to residual connections in ResNets**



Encoder – Decoder hourglass

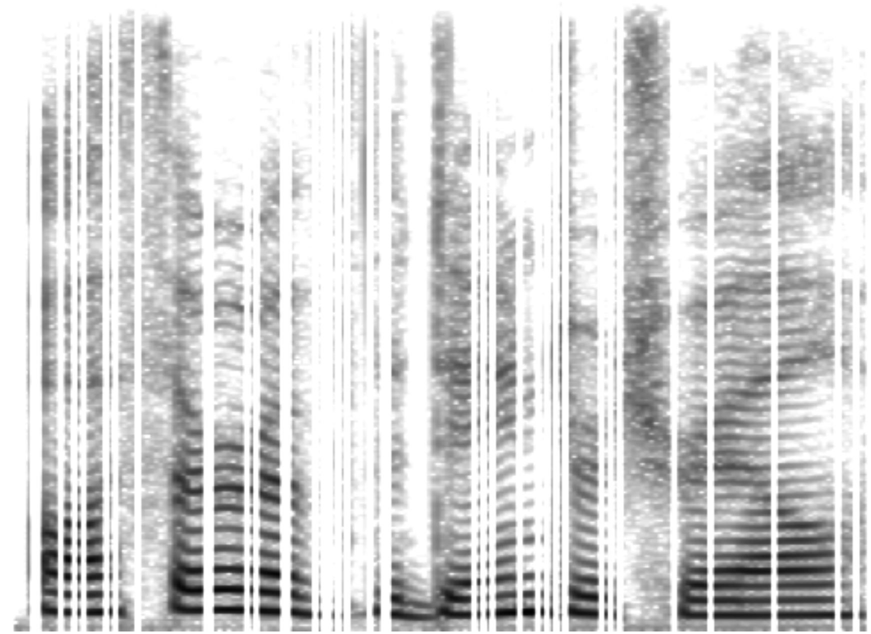
- Common visualisation for autoencoders
- Do the blocks refer to layer sizes or activation map sizes?
- Usually the reduction is exponential (e.g., repeated down/upsample by factor of two)
- Pictures often show a linear reduction



Aalto University
School of Electrical
Engineering

Generative models

- How to handle packet dropout and other corruptions that can not be filtered out?
- Generative models can fill in the gaps with plausible content
- Masked prediction is used as a training technique for GPTs and other generative models



Project: Speech Enhancement

- **Part 1 – Experiments**
  - Implement a denoising neural net model
  - Implement data providers and training
  - Implement metrics
  - Re-use of code from exercises helps
  - Submit code and trained model for evaluation
- **Part 2 – Report**
  - Describe experiments and results

Project timeline

- **Python package template and project specs will be released on Week 9**
- **Milestones based on unit-tests**
  - Milestone 1: DL Week 11
  - Milestone 2: DL Week 13
  - Assemble code from exercises to build a system
  - Probably autograded on JupyterHub
- **Final report deadline 18.4.**
  - Submit a trained model, autograded metrics for bonus points

Lecture 06 - Summary

- Speech denoising and enhancement
- Autoencoders
- Denoising autoencoders
- Data augmentation
- U-Net architecture
- Course project