Deep Learning

Lecture 3: Designing architectures

Chris G. Willcocks

Durham University

Lecture overview

1 Architecture shapes and design

- natural signals
- convolutional neural networks
- encoders (classification and regression)
- autoencoder
- recurrent neural networks
- deep residual networks
- skip connections (U-Net)

2 Regularisation

- early stopping and annealing
- the effect on model capacity
- data augmentation
- dropout and Tikhonov regularization
- ensembles

Natural signals

In nature, the data signal follows patterns:

- There are repetitions in **space** and **time**
- There are various **symmetries**
- Signals are **hierarchical**

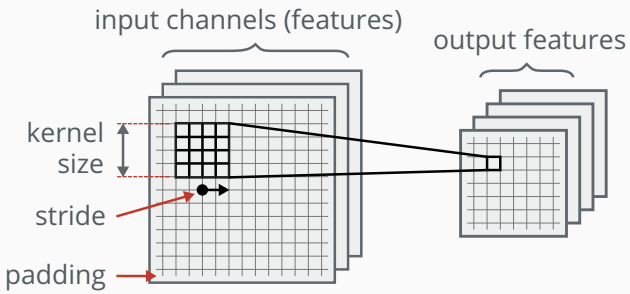
Therefore we can design our functions to fit these patterns effectively

Further reading about deep learning through a physics lens in [1]

Definition: CNNs

A convolution sums the Hadamard product between a sliding area and the convolution kernel. Each output feature map does this for each input channel.

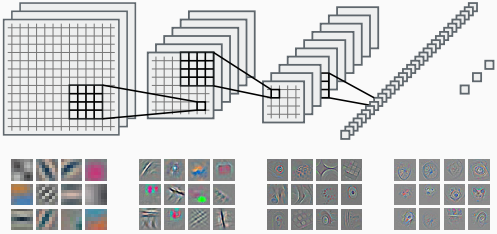
This 2D convolution example is a 4×4 kernel with a stride of 2 and 1 padding. It has 3 input channels and 4 output features.



Hierarchical design

Each feature map output is essentially an image whose intensity values are ‘feature detectors’ from the layer before. The images here are learnt kernel weights.

The ‘width’ vs ‘depth’ problem is based on what type of detectors you need from the dataset, such as to minimise the task risk.



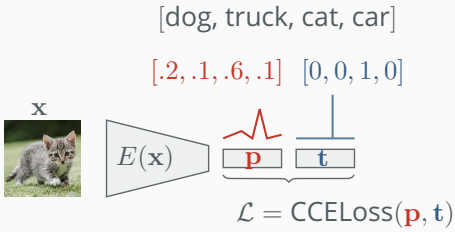
Architecture shapes encoder (classification and regression)

Definition: encoder

Encoders typically reduce the spatial dimensions (compress) but may increase the feature dimensions.

Link to Colab example

- For regression, we generally use an L_2 loss
- For classification, we generally use categorical cross entropy
- For binary classification, you can use binary cross entropy



Architecture shapes autoencoder

Definition: autoencoder

Autoencoders [2] have two parts, an encoder $E(\mathbf{x})$ and a decoder $D(\mathbf{z})$, with the goal of learning an encoding $\mathbf{z}$ (a representation typically of reduced dimension), e.g. by minimising the squared L2 loss:

$$\mathcal{L}_{\text{AE}} = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}\left[\|\mathbf{x} - D(E(\mathbf{x}))\|^2\right]$$

Link to Colab example

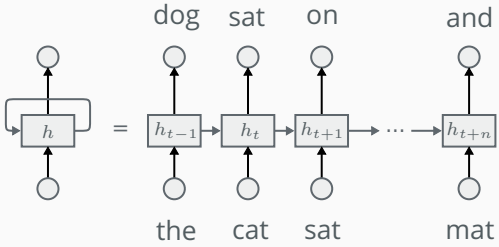
- Can you sample $\mathbf{z} \sim \mathcal{N}$ from a normal distribution?
- What happens if you compress $\mathbf{z}$ to 1D or higher dimensions (more signal bandwidth)?

Reconstructive autoencoder:

Paired translation:

Definition: recurrent neural network

RNNs reuse parameters across multiple timesteps. They can be unrolled to better understand their dynamic behaviour.

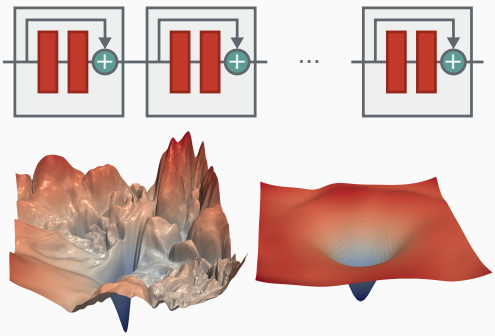


Definition: residual connections

These are shortcuts that skip over two or three layers that contain nonlinearities and batch normalisation between then.

Pseudocode: residual block

```
class ResidualBlock(nn.Module):
  def init(n):
    res_block = [
    nn.Conv2d(in_f=n, out_f=n,3,1,1),
    nn.BatchNorm2d(n),
    nn.ReLU(),
    nn.Conv2d(in_f=n, out_f=n,3,1,1),
    nn.BatchNorm2d(n) ]
  def forward(x):
    return F.relu(x + res_block(x))
```



loss landscape without residuals

loss landscape with residuals [3]

Architecture design skip connections (U-Net)

Definition: skip connections (U-Net)

Skip connections (U-Net) is a popular residual approach used for paired image translation tasks [4]. For example for images $\mathbf{x}$ and paired masks $\mathbf{m}$, where:

$$\mathcal{L} = \mathbb{E}_{\mathbf{x},\mathbf{m}\sim p_{\text{data}}} \left[\|D(E(\mathbf{x})) - \mathbf{m}\|^2 \right]$$

Link to Colab example

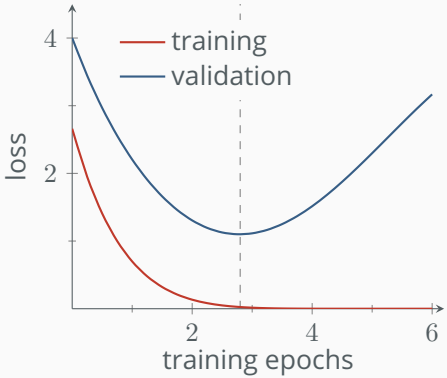
Before we define regularisation, first lets examine two ways to prevent overfitting in high-capacity models:

Definition: early stopping

This is just where we stop training early.

Definition: annealing

If we decrease the learning rate slowly to zero, this has a similar effect to early stopping (but allows more experience in the process).

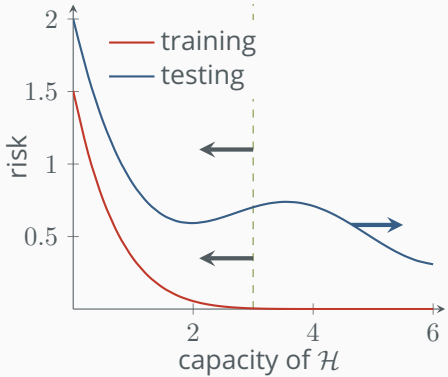


Regularisation the effect on model capacity

Definition: regularisation

Regularisation is where we add prior information about functions in $\mathcal{H}$ such as to reduce generalisation error.

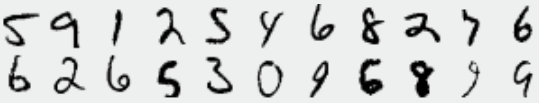
We'll learn more about deep double descent in a later lecture on generalisation theory.



Definition: data augmentation

If it is expected that small transformations (e.g. rotations, zooms, flips, blurs) will occur in testing, the training samples can be augmented.

However too much augmentation (e.g. too much zoom) will result in poor fitting. In the extreme case it may even change the class label, for example 180° rotations in MNIST:



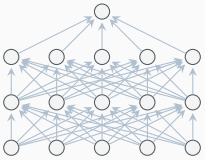
Definition: dropout

Dropout is where each hidden unit is set to zero with some probability (e.g. 0.2). The network can't rely on any one weight, so it spreads its weights out.

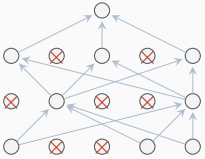
Definition: Tikhonov regularization

Tikhonov regularization (also called **weight decay** or L_2 **regularisation** has a similar effect:

$$L' = L + \lambda ||\mathbf{w}||_2^2$$



standard network

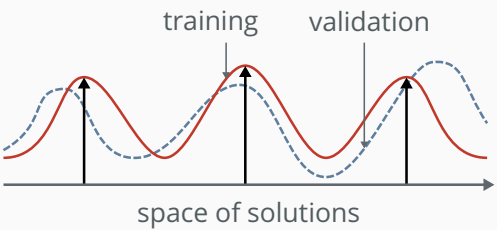


after dropout

Definition: an ensemble

An ensemle is where multiple different models are trained, and then the predictions are combined at test time, for example by averaging or max voting.

This simple technique has shown to be highly successful in winning kaggle competitions, where there is evidence to suggest the success is due to the ability for ensembles to capture multiple modes of the solution space [5].



Take Away Points

Summary

In summary, designing architectures:

- is a scientific process
- design the right shape to transform the data
- choose the right functions to fit the data
- choose your experiments carefully
- what do you know about the signal?
- what does the task need?

References I

[1] Henry W Lin, Max Tegmark, and David Rolnick. “Why does deep and cheap learning work so well?” In: Journal of Statistical Physics 168.6 (2017), pp. 1223–1247.

[2] Mark A Kramer. “Nonlinear principal component analysis using autoassociative neural networks”. In: AIChE journal 37.2 (1991), pp. 233–243.

[3] Hao Li et al. “Visualizing the loss landscape of neural nets”. In: Advances in Neural Information Processing Systems. 2018, pp. 6389–6399.

[4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional networks for biomedical image segmentation”. In: International Conf on Medical image comp and comp-assisted intervention. Springer. 2015, pp. 234–241.

[5] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. “Deep ensembles: A loss landscape perspective”. In: arXiv preprint arXiv:1912.02757 (2019).