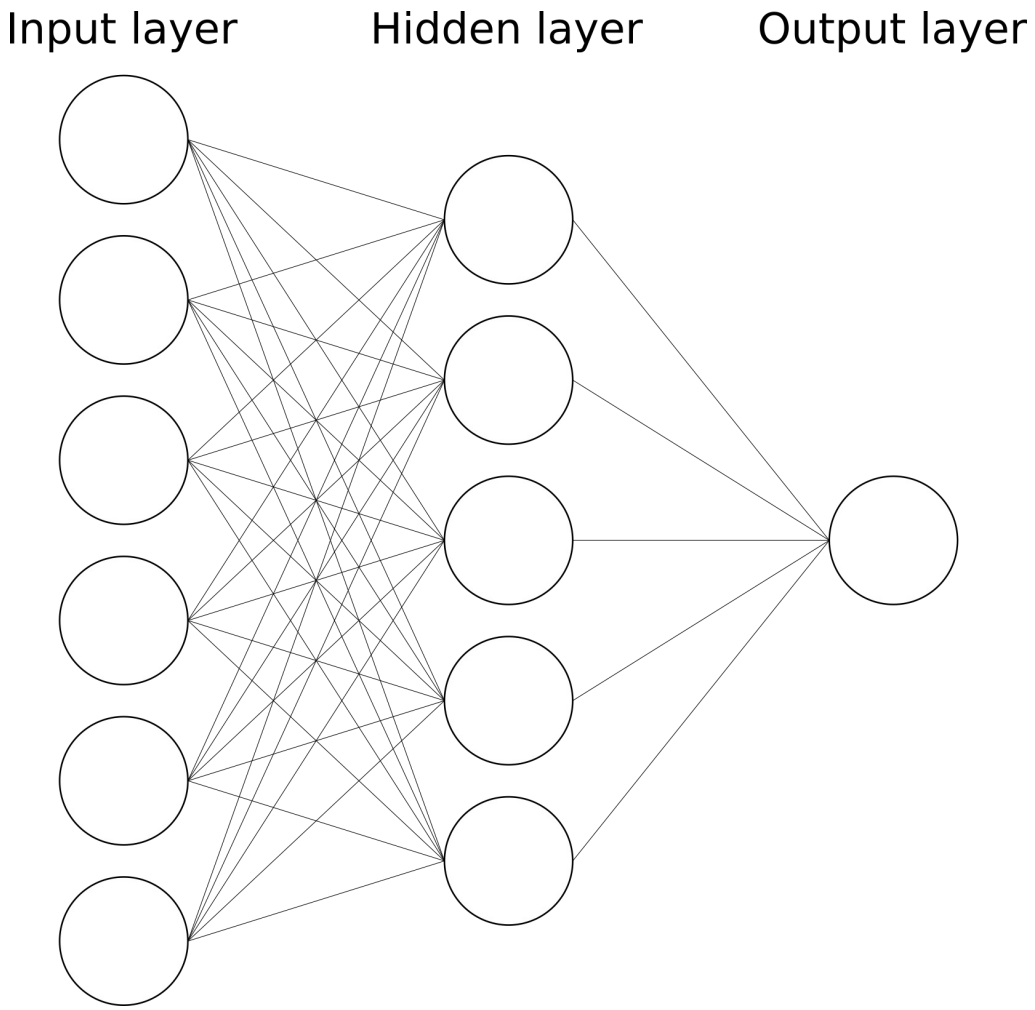
Deep Learning

Jonas Birkelund Nilsson

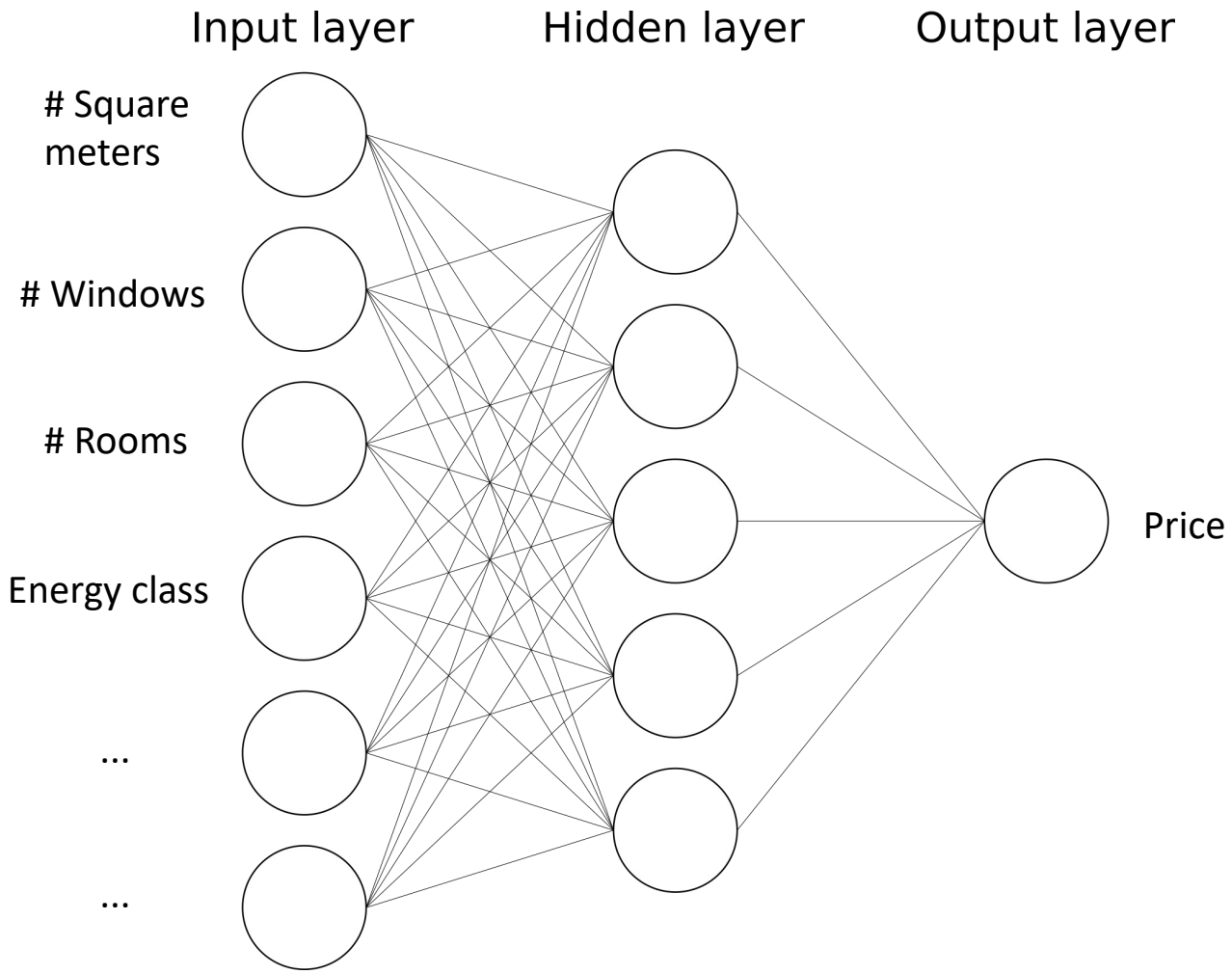
Research Assistant, Immunoinformatics and Machine Learning Group

What we have used so far – Feed-forward neural network (FFN)



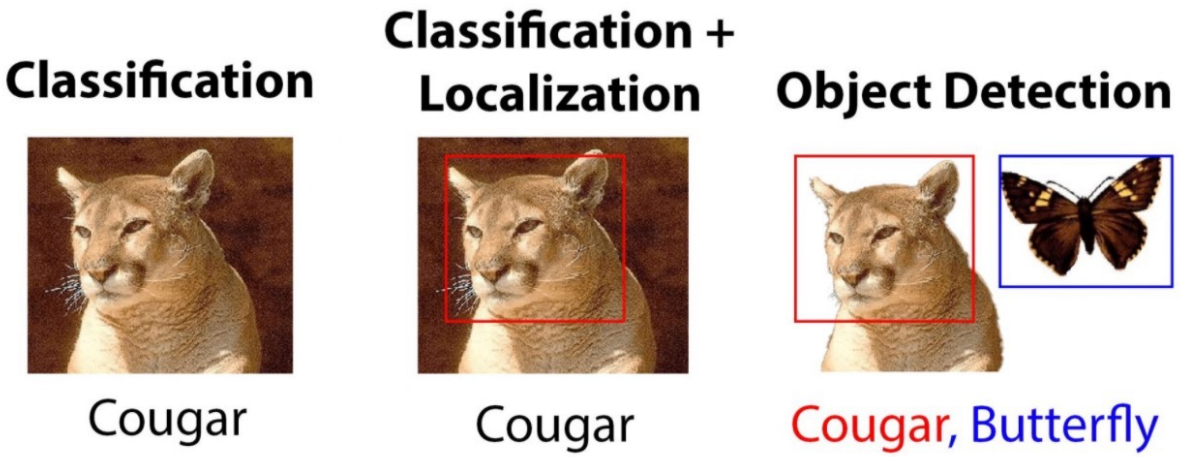
When do FFNs work?

Example:
Predict price of house based on several features

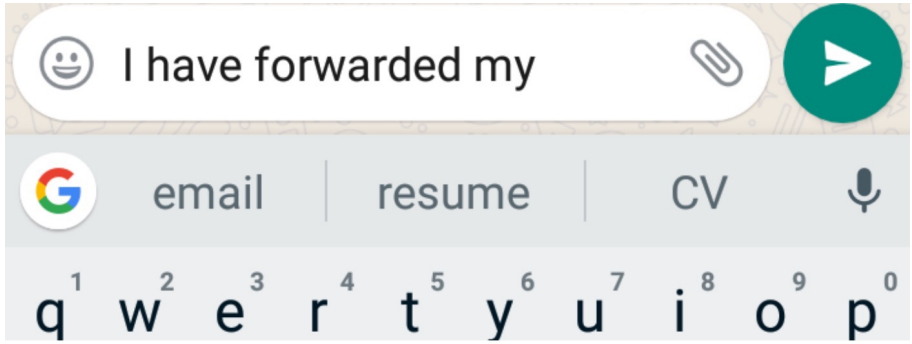


When do FFNs not work?

Image analysis

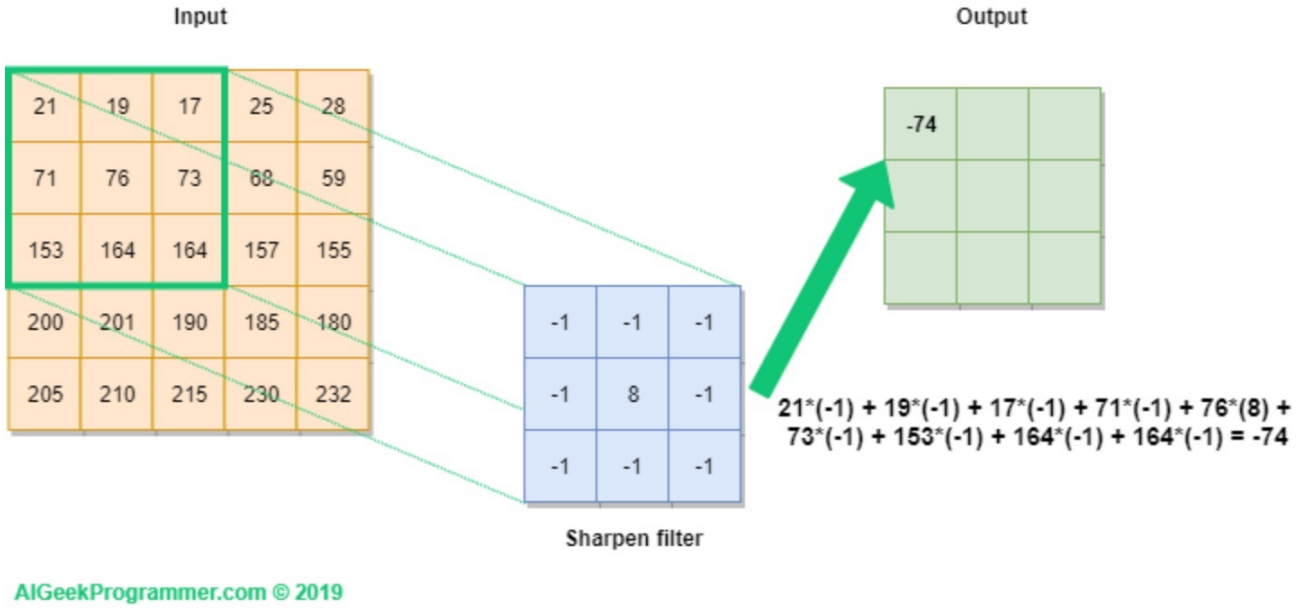


Sequential data



Convolutional Neural Networks (CNNs)

- Developed for images, but they work with other types of input, such as peptide sequences!
- Able to handle inputs with different sizes
- Able to extract the relevant features from the input by themselves



How convolution works

Input vector

0.2 -0.2 0.2 0.0 0.3 -0.1 0.5

Filter

0.1 0.4 -0.2 0.3

Convolutional layer

0.1 0.4 -0.2 0.3
0.2 -0.2 0.2 0.0 0.3 -0.1 0.5 — 0.1·0.2 + 0.4·(-0.2) + (-0.2)·0.2 + 0.3·0.0 = -0.1

0.2 0.1 0.4 -0.2 0.3 -0.1 0.5 — 0.1·(-0.2) + 0.4·0.2+(-0.2)·0.0 + 0.3·0.3 = 0.15

0.2 -0.2 0.1 0.4 -0.2 0.3 0.5 — 0.1·0.2 + 0.4·0.0 + (-0.2)·0.3 + 0.3·(-0.1) = -0.07

0.2 -0.2 0.2 0.1 0.4 -0.2 0.3 — 0.1·0.0 + 0.4·0.3 + (-0.2)·(-0.1) + 0.3·0.5 = 0.29

Output:

-0.1 0.15 -0.07 0.29

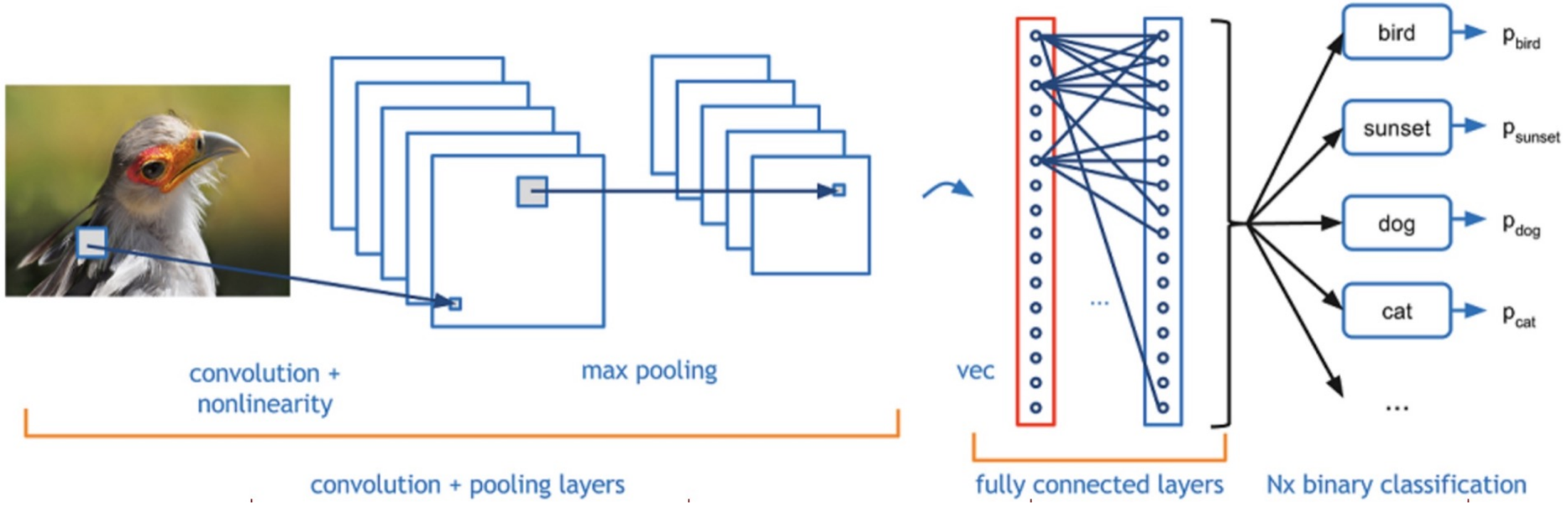
Max pooling:

Forward the highest value

Final output:

0.29

Convolutional Neural Networks



Feature extraction

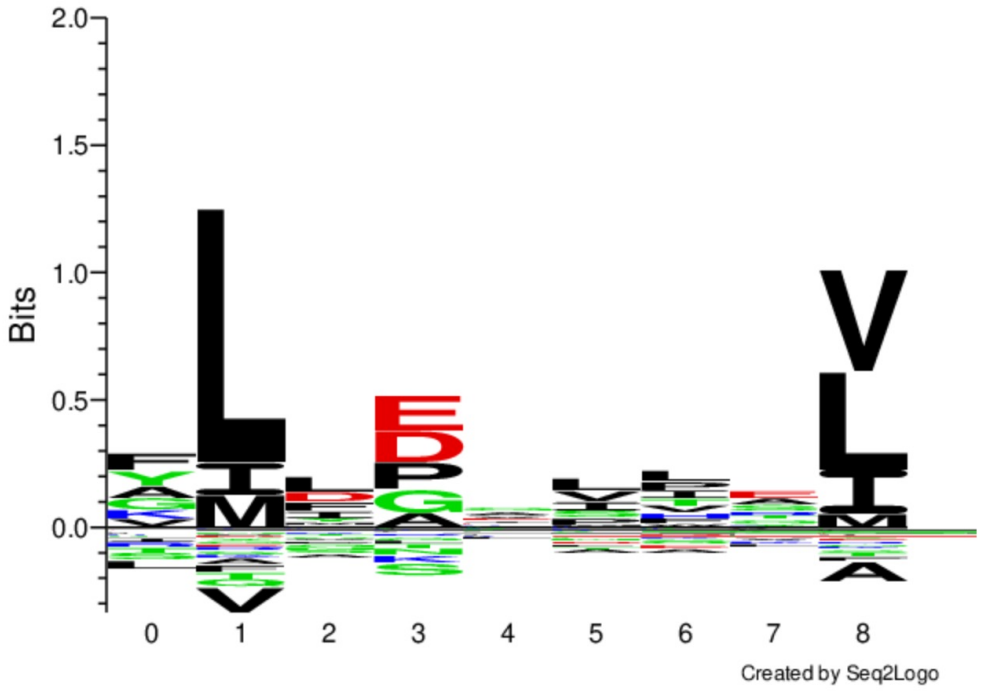
Classification

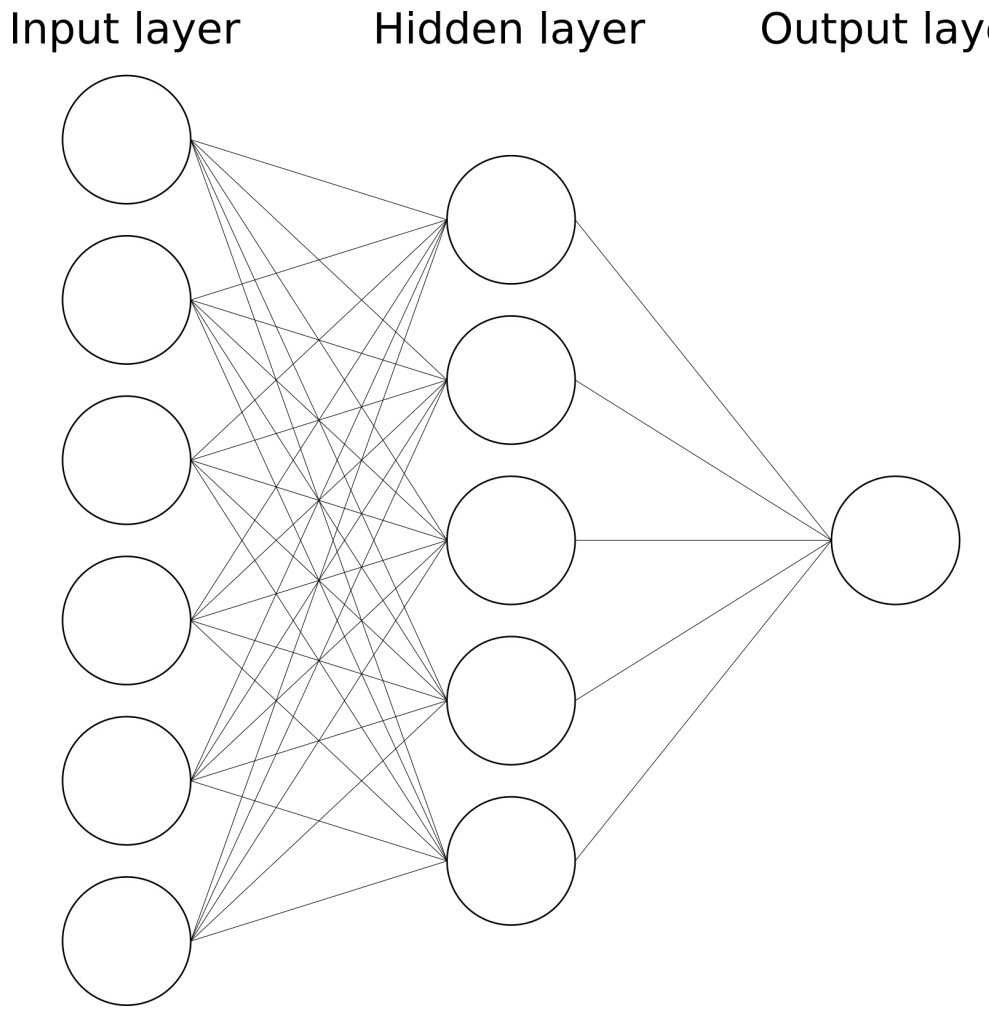
Visualizing the CNN filters

Today’s task: predict peptide-MHC binding

HLA-A*02:01

- Mostly 9-mer peptides bind
- The anchor positions stay ‘fixed’ (P2 and P9)
- We can use a FFN to model this

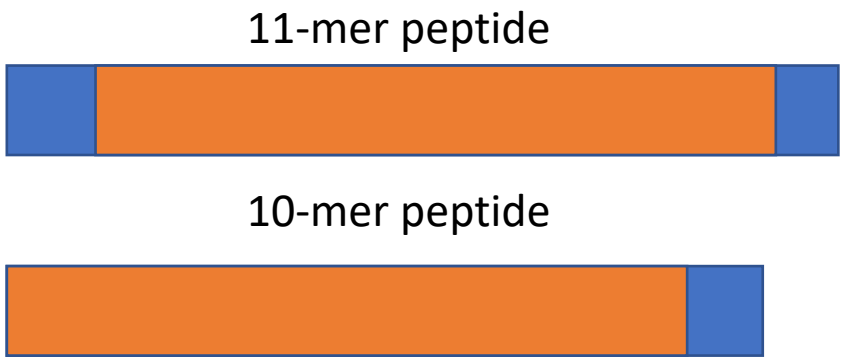


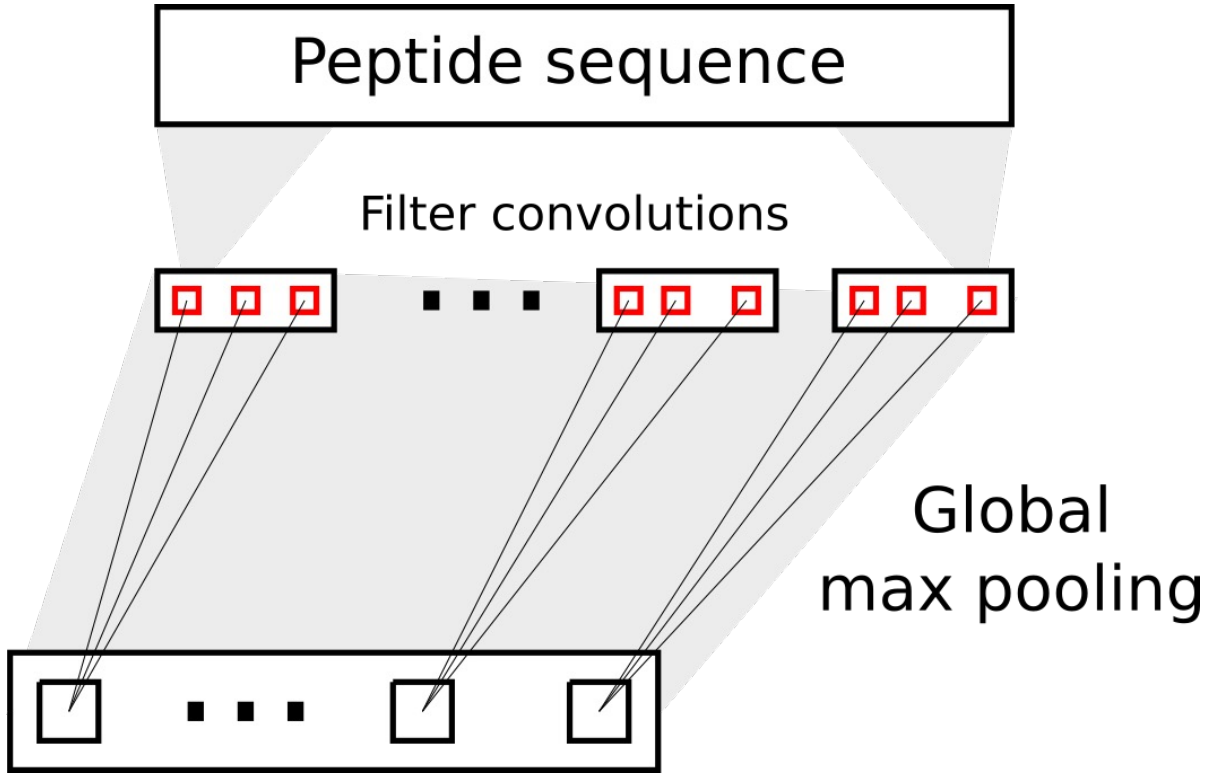


Today’s task: predict peptide-MHC binding

HLA-A*03:01

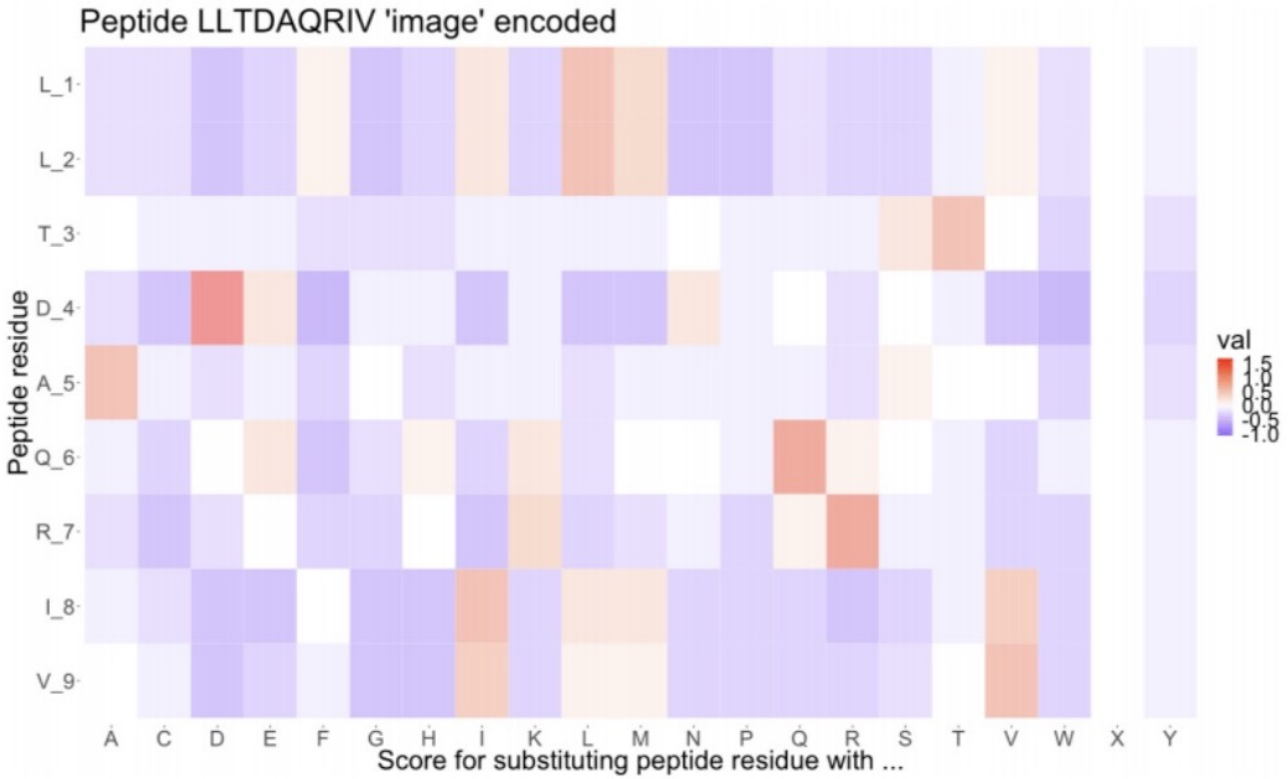
- Peptides of lengths 8,9,10,11
- The binding core is not ‘fixed’ in one spot
- A FFN will most likely fail
- Instead, use a CNN!





Peptide encoding

- The amino acid sequences are converted to numbers based on their biochemical profile (BLOSUM)
- Each AA is represented by a vector with 20 entries
- Example:
  A 9-mer is encoded as a [9x20] matrix



In practice

- Use 1D convolution with 20 channels (due to BLOSUM encoding)
- As such, each filter is a Kx20 weight matrix, where K is the number of amino acids covered by the filter
- Use global max pooling to forward the maximum score from each filter

21 due to the inclusion of 'X' (unknown AA)

```
class CNNpep(nn.Module):

    def __init__(self, n_filters, k, n_l1):
        super(CNNpep, self).__init__()
        self.conv_layer = nn.Conv1d(in_channels=21,
                                    out_channels=n_filters,
                                    kernel_size=k,
                                    stride=1,
                                    padding=0)

        self.fc1 = nn.Linear(n_filters, n_l1)
        self.fc2 = nn.Linear(n_l1, 1)
        self.relu = nn.ReLU()
        self.sigmoid = nn.Sigmoid()

    def forward(self, x):
        # The input dimensions are [batch_size, pep_len, channels]
        # PyTorch wants [batch_size, channels, pep_len]
        # So we swap the second and third dimentions
        x = x.permute(0, 2, 1)
        x = self.relu(self.conv_layer(x))
        x, _ = torch.max(x, axis=2)
        x = self.relu(self.fc1(x))
        out = self.sigmoid(self.fc2(x))

        return out
```

Today’s exercise

- FNN on A0201-restricted peptides
- FNN on A0301-restricted peptides (look at the peptide length distribution)
- CNN on A0301-restricted peptides

Challenge:
Find the optimal performing model

https://docs.google.com/spreadsheets/d/1yEOE9AVc9_m_2QhwNeSSI0uAtGKkn9nQcQFh84HTtOk/edit?usp=sharing

Good luck!