Week 5 – Recurrent NNs

Sequential data – text, video and audio, finance, medicine, etc.

Language Model – generative model of natural language – i.e. p(text) = p(x0, x1, …, xn)

We assume we generate text word-by-word

P(x0)p(x1|x0)p(x2|x1, x0)

MLPs not suitable because of the arbitrary length of sequences. MLPs only work with fixed number of inputs. WHY?

(1)

Could use window – but needs to be long and isn’t always a fixed size

Word embeddings are the input and the y^ is a prob distribution for the next word.

Deals with POS tags.

We only see one input per time step.

(2)

Can use same MLP at every time step meaning that the parameters of the MLP are shared between time steps, meaning a lot fewer parameters.

(3)

Note how the same matrices are used at every time step.

Folded and unfolded forms.

Training:

Sum up losses at each time step to get total loss L = Lt-1 + Lt-1

In the forward pass we calculate losses, hidden elements and our predictions.

We backpropagate through layers and through time.

**Explosion**

If the spectral matrix norm of the Jacobians is greater than 1 then the learning process becomes unstable.

**Vanishing Gradients**

If the spectral matrix norm of the Jacobians is less than 1 then the gradient tends to zero, meaning that contributions from faraway steps vanish and don’t affect training.

(4)

- LSTM, GRU are architectures used to reduce problem.

- ReLu is an effective activation function. Leaky ReLU

- Skipping connections – shortcuts between time steps with their own parameters. When backprop happens we can create longer range connections.

**Gradient Clipping**

If magnitude of the gradient is larger than a threshold then clip:

(5)

(6)

BPTT = back propagation through time. Realistically we truncate instead of running through entire history as this would be too expensive. This will remove the effect of long range dependencies.

**LSTM**

When backp through layers of non-linearity causes vanishing gradient. The main idea to solve this in LSTMs is adding a new separate way through the recurrent layer.

The LCM later has its own internal memory C, which other layers of the network don’t have access to.

At each layer we compute not onlt the vector of hidden units H but also the vector of memory cell C (of the same dimension).

(7)

Input gate controls what to store in memory.

Output gate controls what to get from memory and return to the world.

Gradients do not vanish as there is a short way between ct and ct-1 with no multiplication or non-linearity.

We cannot erase anything in memory cell C and they have finite memory.

To erase stuff we can use a forget gate.

(8)

By choosing suitable bioases LSTM does not forget at first but learns to forget if required.

It as 4 times the amount of paramteres making it less efficient in time and memory and makes overfitting more likely.

**GRU (Gated Recurrent Unit)**

Almost the same quality as LSTM.

(9)

Initialise the bias of the update gate vectors with high values to avoid vanishing gradient.

LSTM: more flexible

GRU: less parameters

Can stack both. Final layer should be LSTM since GRU can’t work with outputs as accurately as LSTM.