Highway Networks

(R. K. Srivastava, Greff, and Schmidhuber 2015)

Standard NN layer: $y = H(x, W_H)$ where H is a non-linear transformation parametrized by weights W_H

Highway network:

$$y = H(x, W_H) \cdot T(x, W_T) + x \cdot C(x, W_C)$$

where T is the *transform* gate and C is the *carry* gate, which define the ratio in which the output is defined by transforming the input in contrast to carrying it over. For simplicity, $C = 1 - T$, producing:

$$y = H(x, W_H) \cdot T(x, W_T) + x \cdot (1 - T(x, W_T))$$

Note: this formulation, where each layer can propagate its input x further requires that all of the elements have the same dimension (y, x, T, H). An option here is to use padding to upscale x or sub-sampling, in order to reduce the dimensionality. An option is also to use a regular layer (without the highway connections) to change the dimensionality, and then continue with the highway layers.

Dropout

Introduced in: (Hinton et al. 2012)

Backpropagation

Introduced in: (Rumelhart, Hinton, and Williams 1985)

Maxout networks

(Goodfellow et al. 2013)

Maxout networks use the *maxout* function as the activation. For an input $x \in \mathbb{R}^d$ the maxout is:

$$h_i(x) = max_{j \in [1,k]} z_{ij}$$

where $z_{ij} = x^T W_{\dots ij} + b_{ij}$, $W \in \mathbb{R}^{d \times m \times k}$ and $b \in \mathbb{R}^{m \times k}$ are the learned model parameters.

Essentially: instead of projecting into the output dimension m, project into $m \times k$ and max over the k additional dimensions. Pytorch impl:

Grid LSTM

(Kalchbrenner, Danihelka, and Graves 2015)

Similar to Multi-dimensional Recurrent Neural Networks (Graves and Schmidhuber 2009)

LSTM along each dimension of network (depth, T). The vertical LSTM hidden / cell states initialized by the inputs.

N-dimensional Grid LSTM accepts N hidden vectors $h_1, \dots, h_N$ and N memory vectors $m_1, \dots, m_N$, which are all distinct for each dimension.

All of the hidden states are then concatenated:

$$H = \begin{bmatrix} \hat{h}_1 \\ \vdots \\ \hat{h}_N \end{bmatrix} \tag{1}$$

The N-dimensional block then computes N LSTM tranfsorms, one for each dimension. Each LSTM transform has its individual weight matrices. Each block accepts input hidden and memory vectors from N dimensions, and outputs them into N dimensions.

$$\begin{aligned} (\hat{h}_1, \hat{m}_1) &= LSTM(H, m_1, W_1) \\ &\dots \\ (\hat{h}_N, \hat{m}_N) &= LSTM(H, m_N, W_N) \end{aligned} \tag{2}$$

CONT

References

Goodfellow, Ian J, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. 2013. "Maxout Networks." *arXiv Preprint arXiv:1302.4389*.

Graves, Alex, and Jürgen Schmidhuber. 2009. "Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks." In *Advances in Neural*

Information Processing Systems, 545–52.

Hinton, Geoffrey E, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. "Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors." *arXiv Preprint arXiv:1207.0580*.

Kalchbrenner, Nal, Ivo Danihelka, and Alex Graves. 2015. "Grid Long Short-Term Memory." *arXiv Preprint arXiv:1507.01526*.

Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams. 1985. "Learning Internal Representations by Error Propagation." California Univ San Diego La Jolla Inst for Cognitive Science.

Srivastava, Rupesh Kumar, Klaus Greff, and Jürgen Schmidhuber. 2015. "Highway Networks." *arXiv Preprint arXiv:1505.00387*.