

Layers

- [BatchNorm](#)
- [Convolution](#)
- [Dropout](#)
- [Linear](#)
- [LSTM](#)
- [Pooling](#)
- [RNN](#)

BatchNorm

BatchNorm accelerates convergence by reducing internal covariate shift inside each batch. If the individual observations in the batch are widely different, the gradient updates will be choppy and take longer to converge.

The batch norm layer normalizes the incoming activations and outputs a new batch where the mean equals 0 and standard deviation equals 1. It subtracts the mean and divides by the standard deviation of the batch.

Code

Code example from [Agustinus Kristiadi](#)

```

def BatchNorm():
    # From https://wiseodd.github.io/techblog/2016/07/04/batchnorm/
    # TODO: Add doctring for variable names. Add momentum to init.
    def __init__(self):
        pass

    def forward(self, X, gamma, beta):
        mu = np.mean(X, axis=0)
        var = np.var(X, axis=0)

        X_norm = (X - mu) / np.sqrt(var + 1e-8)
        out = gamma * X_norm + beta

        cache = (X, X_norm, mu, var, gamma, beta)

        return out, cache, mu, var

    def backward(self, dout, cache):
        X, X_norm, mu, var, gamma, beta = cache

        N, D = X.shape

        X_mu = X - mu
        std_inv = 1. / np.sqrt(var + 1e-8)

        dX_norm = dout * gamma
        dvar = np.sum(dX_norm * X_mu, axis=0) * -.5 * std_inv**3
        dmua = np.sum(dX_norm * -std_inv, axis=0) + dvar * np.mean(-2. * X_mu, axis=0)

        dX = (dX_norm * std_inv) + (dvar * 2 * X_mu / N) + (dmua / N)
        dgamma = np.sum(dout * X_norm, axis=0)
        dbeta = np.sum(dout, axis=0)

        return dX, dgamma, dbeta

```

Further reading

- [Original Paper](#)
- [Implementing BatchNorm in Neural Net](#)
- [Understanding the backward pass through Batch Norm](#)

Convolution

Be the first to [contribute!](#)

Dropout

Be the first to [contribute!](#)

Linear

Be the first to [contribute!](#)

LSTM

Be the first to [contribute!](#)

Pooling

Max and average pooling layers.

Be the first to [contribute!](#)

RNN

Be the first to [contribute!](#)

References

[1] <http://www.deeplearningbook.org/contents/convnets.html>