MLP (Multilayer Perceptron)

$\hat{y} = g(W h(V x)) = g(W z)$

Regression identity function + L2 loss : Gaussian likelihood.

$\hat{y} = g(Wz) = Wz$

$L(y, \hat{y}) = \frac{1}{2} \|y - \hat{y}\|_2^2 = \log N(y; \hat{y}, \beta I) + \text{constant}.$

Binary classification logistic sigmoid + CE : Bernoulli

$\hat{y} = g(Wz) = (1 + e^{-Wz})^{-1}$

$L(y, \hat{y}) = y \log \hat{y} + (1-y) \log (1 - \hat{y}) = \log \text{Bernoulli}(y; \hat{y})$

Multiclass classification softmax + multi-class CE : Categorical

$\hat{y} = g(Wz) = \text{softmax}(Wz)$

$L(y, \hat{y}) = \sum_k y_k \log \hat{y}_k = \log \text{Categorical}(y, \hat{y})$

Activation function (for middle layers)

Universal approximation problem

An MLP with single hidden layer can approximate any continuous function with arbitrary accuracy.

Note: Only about training error.

Deep network (with ReLU activation) is also universal

(Increasing depth is more effective empirically)