

# SwiftAI: Lesson 2 - Fully connected models

$Y \in N \times M, W \in D \times M, X \in N \times D, L \in M \times M$

$$L = (XW - Y)^T (XW - Y) = (U - Y)^T (U - Y) = U^T \Sigma U + Y^T \Sigma Y - 2U^T \Sigma Y$$

$(\Sigma = I)$

$$\frac{\partial L}{\partial U} \frac{\partial U}{\partial X} = \underbrace{(2U - 2Y)}_{N \times M} \underbrace{(W^T)}_{M \times D} = \frac{\partial L}{\partial Y} W^T = U^T U + Y^T Y - 2U^T Y$$

$$\frac{\partial L}{\partial Y} = 2Y - 2U = 2(Y - XW)$$

$N \times M$

$$\frac{\partial L}{\partial U} \frac{\partial U}{\partial W} = \frac{\partial L}{\partial W} = \underbrace{(2U - 2Y)}_{N \times M} X = X^T \frac{\partial L}{\partial Y} \quad (w \text{ grad})$$

$D \times M \quad N \times M$

$$\frac{\partial Y}{\partial W} = \frac{\partial Y}{\partial U} \frac{\partial U}{\partial W} = 1^T \frac{\partial U}{\partial W} = 1^T X \quad (b. \text{ grad})$$

Let  $U = XW$ , and  $\frac{\partial L}{\partial U} = -\frac{\partial L}{\partial Y} = G$

$$\delta L = G : dU$$

$$= G : dX \cdot W + G : X \, dW$$

$$= G W^T : dX + X^T G : dW$$

$(A : B = \text{Tr}(A^T B))$

how?  $G^T dX W = W G^T dX ?$

$$\frac{\partial L}{\partial X} = G W^T \quad \frac{\partial L}{\partial W} = X^T G$$

Non-idiomatic manual gradient

```
class TFGrad {
  var inner, grad: TF
  init (- x: TF) {
    .inner = x
    grad = TF(zeros: x.shape)
  }
}

func lin(- x: TFGrad, - w: TFGrad, - b: TFGrad) -> TFGrad {
  return TFGrad(x.inner * w.inner + b.inner)
}
```