(Q2)

$L(\hat{y}, y) = -y \ln(\hat{y}) - (1-y)\ln(1-\hat{y})$

2.1) $\hat{y}\nabla L(\hat{y}, y) = -\frac{y}{(\hat{y})} + \frac{(1-y)}{(1-\hat{y})}$

$= \frac{-y(1-\hat{y}) + \hat{y}(1-y)}{\hat{y}(1-\hat{y})}$

$= \frac{-y + y\hat{y} + \hat{y} - y\hat{y}}{\hat{y}(1-\hat{y})} = \frac{\hat{y} - y}{\hat{y}(1-\hat{y})} = \boxed{\frac{h_2 - y}{h_2(1-h_2)}} = g$

2.2) $\nabla_{a^2} J = g \odot f'(a^{(k)})$

$= \frac{h_2 - y}{h_2(1-h_2)} \cdot h_2(1-h_2)$

$= \boxed{h_2 - y} = g$

$f'(a_2) = f(a_2)(1 - f(a_2)) = h_2 \cdot (1 - h_2)$

↳ $f(x)$ is sigmoid function

$f(a_2) = h_2$

2.3) $\nabla_b{}^2 J = g + \lambda \nabla_{W^2} \Omega(\theta)$ ($\nabla_{W^2}\Omega(\theta) \to 0$) $\to \Omega$ is zero

$\boxed{\nabla_b{}^2 J = h_2 - y} = g$

2.4) $\nabla_{W^2} J = g \cdot h^{(k-1)T} + 0 \quad \to k=2$

$= (h_2 - y) h^{(1)T}$

2.5) $\nabla h^1 J = W^{(k)T} \cdot g = W^{2T}(h_2 - y) \to$ new g

2.6) $\nabla_{b^1} J = g \cdot f(a_1)(1 - f(a_1))$

$= W^{2T} \cdot (h_2 - y) \odot h_1(1 - h_1)$ $\Rightarrow$ new g

$\to k=1$

$\nabla_{W_1} J = g \cdot h^{(k-1)T} + \nabla_{W^k} \Omega(\theta) \longrightarrow W^{2T}(h_2 - y) h_1(1-h_1) \cdot h_0^T$

2.7) Backpropagation would be much slower w/o forward pass. As beginning weights have the most function implemented like $g(h(f(k(x))))$ So getting weights requires a lot of computations in Chain rule Forward pass allows you to compute gradient from the end, so gradient calculation considers only previous calculated gradient, which is much faster.