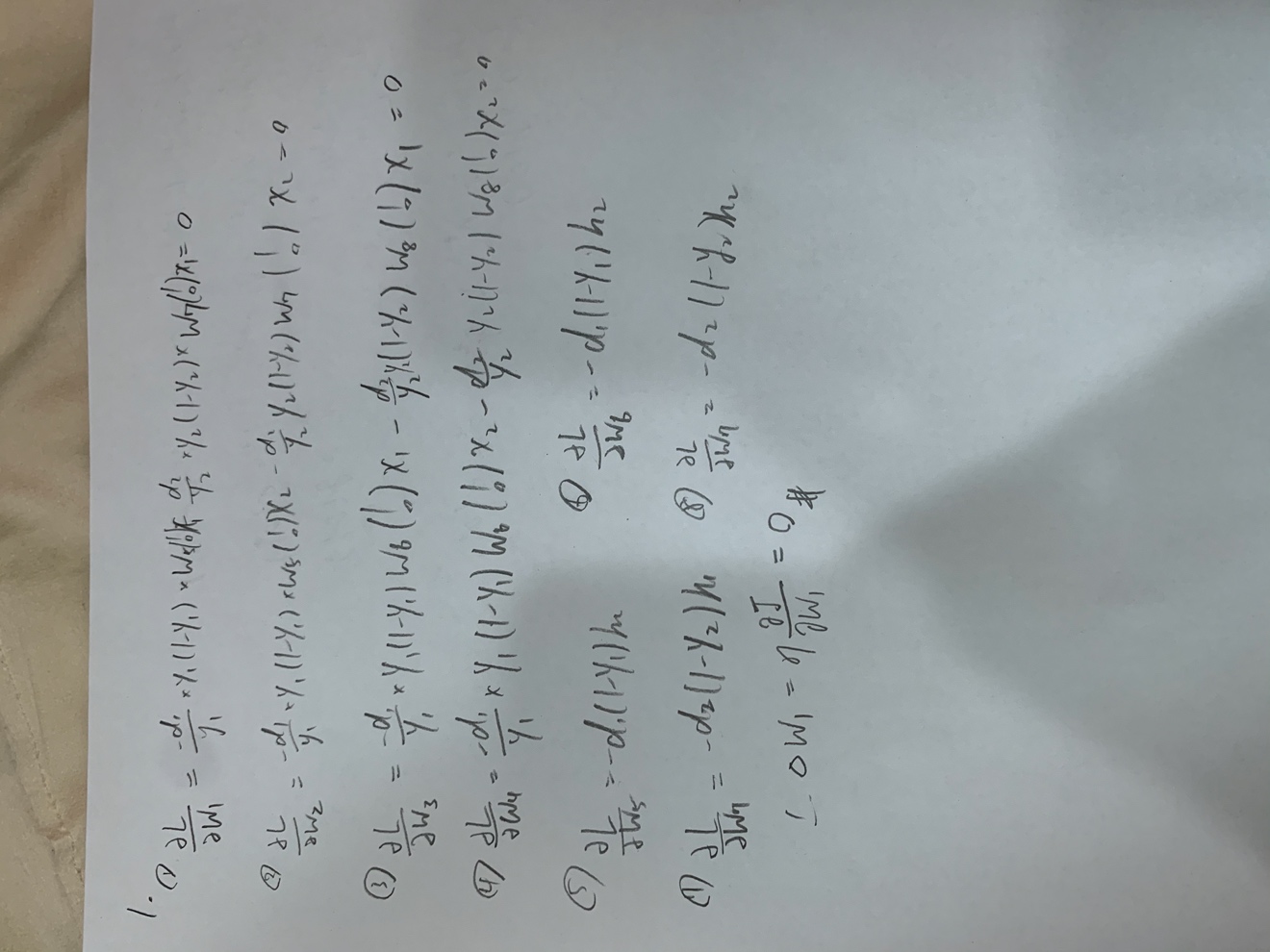
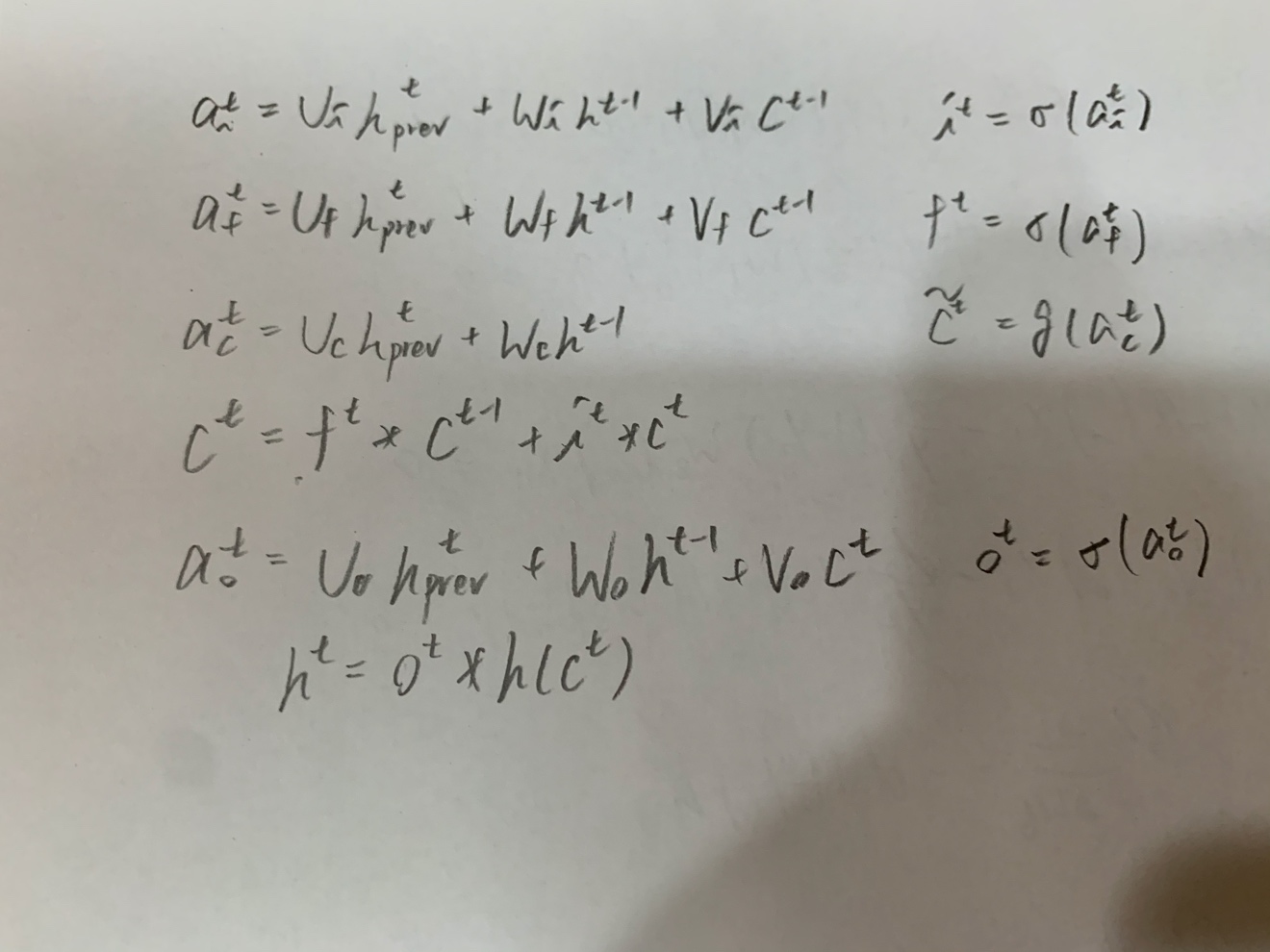
1. **Justify the following claim: The back propagation algorithm does NOT work if weights in the neural network are all set to zero initially. To simplify the problem, use the following network and show that Δ𝑤1 = 𝜂 𝜕𝐽 = 0. The 𝜕𝑤1 activation function from 𝑞1 to h1 and 𝑞2 to h2 is ReLU, the outputs 𝑦1 and 𝑦2 are softmax ouput, and the cost function is 𝐽 = − log 𝑦1. Let 𝑤1 to 𝑤8 be all zeros, 𝑥1 = 0.5, 𝑥2 = 0.5, and 𝜂 = 0.1.**

****

**3. In the version 1 model of the LSTM, the activation function for 𝑓𝑡 is sigmoid (represented by σ in the notes). If we use ReLU instead, can the LSTM nodes still properly work? Why or why not?**

Ans:

No, if we us RELU inside the LSTM, the model would diverge.



The logistic sigmoid is necessary since they serve as “gates” and thus must be bounded in the interval [0, 1]. For the g(.) and h(.) activation functions, we are computing a hadamard product with their output and the output of something else (in different equations). The input gate’s output is element-wise multiplied with the previous context/cell output. The output gate (o) is also element-wise multiplied by the current cell output (C­­t) run through the function h(.) When you have hadamard products, activations can quickly explode if you use activation functions that are not bounded like RELU (since it is unbounded on its positive part).

**4. Assuming that you are designing an anti-spam program based on neural networks with gradient descent. Though the training error seems acceptable, unfortunately the test error is too large to accept. What of the following approaches are appropriate in your case? Explain.**

* Try getting more training examples.
* Try a smaller set of features.
* Try a larger set of features.
* Run gradient descent with a different optimizer (such as RMSprop or ADAM).
* Try Newton’s method.
* Use a smaller value for λ in the L2 norm term.
* Try using an SVM (different type of classifier).

Ans:

1. “Try getting more training examples”: Through getting more training examples, the model might be able to be more general, and we can decrease the test errors then.
2. “Try a smaller set of features”: Since we get large test error, which might mean the model is overfitting; as the result, we can try to decrease the complexity of the model through a smaller set of features, which might help to fix the overfitting problem.
3. “Run gradient descent with a different optimizer”: Since we’re suffering high test errors, trying with a different optimizer might help to decrease the variance.
4. “Try Newton’s method”: Through Newton’s method, we might be able to get a better optimization result, which could help to decrease test errors.
5. “Try using an SVM”: SVM could help to decrease the complexity of the model, which could help with decreasing test errors.