Consider the medical diagnosis problem of classifying tumors as malignant or benign. If a hypothesis $h_\theta(x)$ has overfit the training set, it means that:

- ○ It makes accurate predictions for examples in the training set and generalizes well to make accurate predictions on new, previously unseen examples.
- ○ It does not make accurate predictions for examples in the training set, but it does generalize well to make accurate predictions on new, previously unseen examples.
- ● It makes accurate predictions for examples in the training set, but it does not generalize well to make accurate predictions on new, previously unseen examples.
- ○ It does not make accurate predictions for examples in the training set and does not generalize well to make accurate predictions on new, previously unseen examples.

Consider the medical diagnosis problem of classifying tumors as malignant or benign. If a hypothesis $h_\theta(x)$ has overfit the training set, it means that:

- [ ] It makes accurate predictions for examples in the training set and generalizes well to make accurate predictions on new, previously unseen examples.
- [ ] It does not make accurate predictions for examples in the training set, but it does generalize well to make accurate predictions on new, previously unseen examples.
- [x] It makes accurate predictions for examples in the training set, but it does not generalize well to make accurate predictions on new, previously unseen examples.

Correct

- [ ] It does not make accurate predictions for examples in the training set and does not generalize well to make accurate predictions on new, previously unseen examples.

In regularized linear regression, we choose θ to minimize:

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$

What if λ is set to an extremely large value (perhaps too large for our problem, say $\lambda = 10^{10}$)?

- ○ Algorithm works fine; setting λ to be very large can't hurt it.
- ○ Algorithm fails to eliminate overfitting.
- ◉ Algorithm results in underfitting (fails to fit even the training set).

Correct

- ○ Gradient descent will fail to converge.

Suppose you are doing gradient descent on a training set of $m > 0$ examples, using a fairly small learning rate $\alpha > 0$ and some regularization parameter $\lambda > 0$. Consider the update rule:

$$\theta_j := \theta_j(1 - \alpha\frac{\lambda}{m}) - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}.$$

Which of the following statements about the term $(1 - \alpha\frac{\lambda}{m})$ must be true?

- $1 - \alpha\frac{\lambda}{m} > 1$
- $1 - \alpha\frac{\lambda}{m} = 1$
- $1 - \alpha\frac{\lambda}{m} < 1$

Correct

- None of the above.

When using regularized logistic regression, which of these is the best way to monitor whether gradient descent is working correctly?

- Plot $-[\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))]$ as a function of the number of iterations and make sure it's decreasing.
- Plot $-[\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))] - \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$ as a function of the number of iterations and make sure it's decreasing.
- Plot $-[\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$ as a function of the number of iterations and make sure it's decreasing.

Correct

- Plot $\sum_{j=1}^{n}\theta_j^2$ as a function of the number of iterations and make sure it's decreasing.

Continue