$$\mathcal{L} = x_1^2 + x_2^2 - 1 + \lambda_1 (x_1 + x_2 - 1) + \lambda_2 (2x_2 - x_1)$$

$$\frac{\partial \mathcal{L}}{\partial x_1} = 2x_1 + \lambda_1 - \lambda_2 = 0 \quad \text{——} \quad ①$$

$$\frac{\partial \mathcal{L}}{\partial x_2} = 2x_2 + \lambda_1 + 2\lambda_2 = 0 \quad \text{——} \quad ②$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_1} = x_1 + x_2 - 1 = 0 \quad \text{——} \quad ③$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_2} = 2x_2 - x_1 = 0 \quad \text{——} \quad ④$$

~~①②③④~~

$$2(x_2 - x_1) + 3\lambda_2 = 0$$

$$-2x_2 + 3\lambda_2 = 0$$

~~②~~

$$\lambda_2 = \frac{2}{3} x_2$$

$$2(x_2 - x_1) + 2x_2 = 0$$

$$2x_2 = x_1$$

$$x_1 + x_2 - 1 = 0$$

$$3x_2 = 1$$

$$x_2 = 1/3 \quad //$$

$$x_1 = 2/3 \quad //$$

（科目：Q1.2）**数 学 作 业 纸**

编号：2018280351 班级： 姓名：Nanfu 第 页

Let X denote the length...

Flips	Probability	Length
T	$\frac{1}{2}$	$E(X)+1$
HH	$\frac{1}{4}$	$E(X)$
HTH	$\frac{1}{8}$	$E(X)+1$
$H\underbrace{T\dots T}_{k-1}H$	$\frac{1}{2^{k+1}}$	$E(X)+(k-1)$
$H\underbrace{T\dots T}_{k}$	$\frac{1}{2^{k+1}}$	$k+1$

$$E(X)=\frac{1}{2}(E(X)+1)+\frac{1}{4}E(X)+\frac{1}{8}(E(X)+1)+\dots+\frac{1}{2^{k+1}}\left(E(X)+(k-1)\right)+\frac{1}{2^{k+1}}(k+1)$$

$$\frac{1}{2^{k+1}}\left(E(X)\right)=\frac{1}{2}+\frac{1}{8}+\frac{2}{16}+\dots+\frac{k-1}{2^{k+1}}+\frac{k+1}{2^{k+1}}$$

$$=\frac{1}{2}+\frac{1}{4}\sum_{i=0}^{k-1}\frac{i}{2^i}+\frac{k+1}{2^{k+1}} \quad \text{——— (1)}$$

$$S=\sum_{i=0}^{k-1}\frac{i}{2^i}=\frac{0}{1}+\frac{1}{2}+\frac{2}{4}+\frac{3}{8}+\dots+\frac{k-1}{2^{k-1}}$$

$$\frac{1}{2}S=\frac{1}{4}+\frac{2}{8}+\frac{3}{16}+\dots+\frac{k-1}{2^k}$$

$$\left(1-\frac{1}{2}\right)S=\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\dots+\frac{1}{2^{k-1}}-\frac{k-1}{2^k}$$

$$\frac{1}{2}S = \frac{\frac{1}{2} - \frac{1}{2^k}}{1 - \frac{1}{2}} - \frac{k-1}{2^k}$$

$$= 1 - \frac{1}{2^{k-1}} - \frac{k-1}{2^k}$$

$$= 1 - \frac{k+1}{2^k}$$

$$\sum_{i=0}^{k-1} \frac{i}{2^i} = S = 2 - \frac{k+1}{2^{k-1}} \qquad \text{(2)}$$

Substitute (2) into (1)

$$\frac{1}{2^{k+1}} \quad E(X) = \frac{1}{2} + \frac{1}{4} \sum_{i=0}^{k-1} \frac{i}{2^i} + \frac{k+1}{2^{k+1}}$$

$$= \frac{1}{2} + \frac{1}{4}\left(2 - \frac{k+1}{2^{k-1}}\right) + \frac{k+1}{2^{k+1}}$$

$$= \frac{1}{2} + \frac{1}{2} - \frac{k+1}{2^{k+1}} + \frac{k+1}{2^{k+1}}$$

$$E(X) = 2^{k+1} \;//$$

Rewriting the constraints:

$$\min_{w,b,\xi,\hat{\xi}} \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \hat{\xi}_i)$$

$$\text{s.t.} \quad y_i - w^T x_i - b - \epsilon - \xi_i \le 0 \qquad i = 1 \dots N$$

$$-y_i + w^T x_i + b - \epsilon - \hat{\xi}_i \le 0 \qquad i = 1 \dots N$$

$$-\xi_i \le 0 \qquad i = 1 \dots N$$

$$-\hat{\xi}_i \le 0 \qquad i = 1 \dots N$$

$$\mathcal{L} = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \hat{\xi}_i) + \sum_{i=1}^{N}\alpha_i(-\xi_i) + \sum_{i=1}^{N}\hat{\alpha}_i(-\hat{\xi}_i)$$
$$+\sum_{i=1}^{N}\beta_i(y_i - w^T x_i - b - \epsilon - \xi_i) + \sum_{i=1}^{N}\hat{\beta}_i(-y_i + w^T x_i + b - \epsilon - \hat{\xi}_i)$$

$$\mathcal{L} = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \hat{\xi}_i) - \sum_{i=1}^{N}(\alpha_i\xi_i + \hat{\alpha}_i\hat{\xi}_i)$$
$$+\sum_{i=1}^{N}\beta_i(y_i - w^T x_i - b - \epsilon - \xi_i) + \sum_{i=1}^{N}\hat{\beta}_i(-y_i + w^T x_i + b - \epsilon - \hat{\xi}_i)$$

$$\frac{\partial \mathcal{L}}{\partial w} = w - \sum_{i=1}^{N}\beta_i x_i + \sum_{i=1}^{N}\hat{\beta}_i x_i = 0$$

$$w = \sum_{i=1}^{N}(\beta_i - \hat{\beta}_i)x_i$$

$$\frac{\partial \mathcal{L}}{\partial b} = -\sum_{i=1}^{N}\beta_i + \sum_{i=1}^{N}\hat{\beta}_i = 0$$

$$\sum_{i=1}^{N}(\beta_i - \hat{\beta}_i) = 0$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i} = C - \alpha_i - \beta_i = 0 \qquad \forall\ i = 1 \dots N$$

$$\alpha_i + \beta_i = C$$

$$\frac{\partial \mathcal{L}}{\partial \hat{\xi}_i} = C - \hat{\alpha}_i - \hat{\beta}_i = 0$$

$$\hat{\alpha}_i + \hat{\beta}_i = C \qquad \forall\ i = 1 \dots N$$

Rewriting Primal problem in Dual form given the above conditions:

$$\mathcal{L} = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \hat{\xi}_i) - \sum_{i=1}^{N}(\alpha_i\xi_i + \hat{\alpha}_i\hat{\xi}_i)$$
$$+ \sum_{i=1}^{N}\beta_i(y_i - w^T x_i - b - \epsilon - \xi_i) + \sum_{i=1}^{N}\hat{\beta}_i(-y_i + w^T x_i + b - \epsilon - \hat{\xi}_i)$$

$$= \frac{1}{2}\left(x_i^T(\beta_i - \hat{\beta}_i)^T(\beta_i - \hat{\beta}_i)x_i\right) +$$

$$= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\beta_i - \hat{\beta}_i)(\beta_j - \hat{\beta}_j)x_i x_j +$$

$$= \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i + C\sum_{i=1}^{N}\hat{\xi}_i - \sum_{i=1}^{N}(\alpha_i + \beta_i)\xi_i - \sum_{i=1}^{N}(\hat{\alpha}_i + \hat{\beta}_i)\hat{\xi}_i$$
$$+ \sum_{i=1}^{N}(\beta_i - \hat{\beta}_i)y_i - \sum_{i=1}^{N}(\beta_i - \hat{\beta}_i)w^T x_i - b\sum_{i=1}^{N}(\beta_i - \hat{\beta}_i) - \epsilon\sum_{i=1}^{N}(\beta_i + \hat{\beta}_i)$$

$$= -\frac{1}{2}\|w\|^2 + \sum_{i=1}^{N}(\beta_i - \hat{\beta}_i)y_i - \epsilon\sum_{i=1}^{N}(\beta_i + \hat{\beta}_i)$$

$$\mathcal{L}_D = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\beta_i - \hat{\beta}_i)(\beta_j - \hat{\beta}_j)x_i x_j + \sum_{i=1}^{N}(\beta_i - \hat{\beta}_i)y_i - \epsilon\sum_{i=1}^{N}(\beta_i + \hat{\beta}_i)$$

st $\alpha_i \geq 0$ $\quad \alpha_i + \beta_i = C$

$\hat{\alpha}_i \geq 0$ $\quad \hat{\alpha}_i + \hat{\beta}_i = C$

$\beta_i \geq 0$ $\quad \sum_{i=1}^{N} (\beta_i - \hat{\beta}_i) = 0$

$\hat{\beta}_i \geq 0$

$\Rightarrow \quad 0 \leq \alpha_i \leq C$

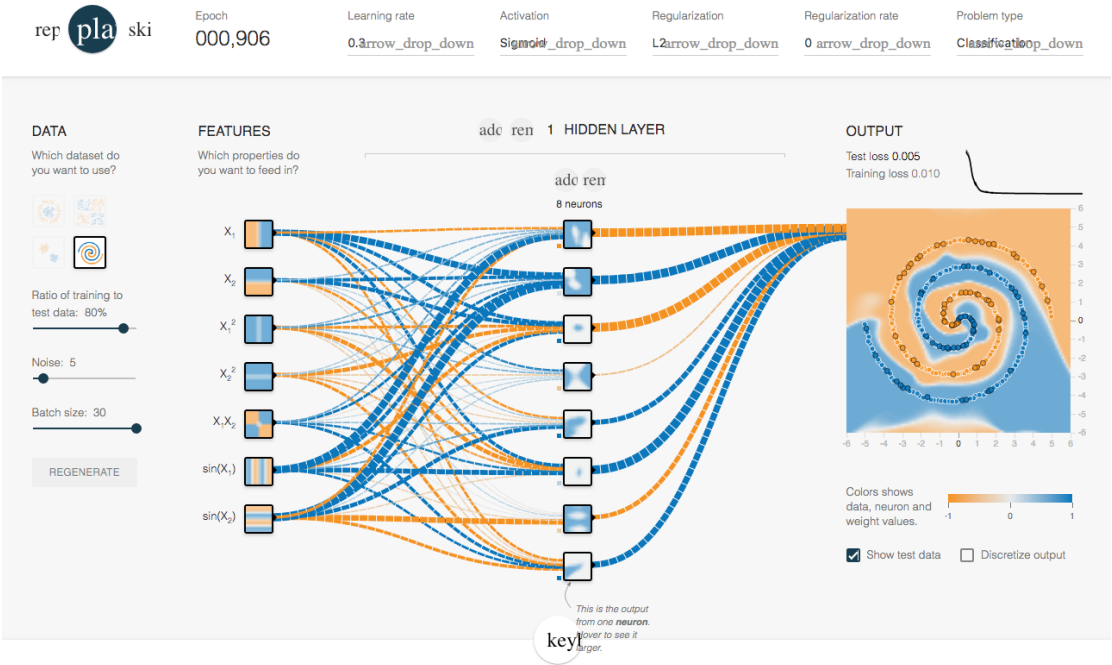
$0 \leq \hat{\alpha}_i \leq C$

$\sum_{i=1}^{N} (\beta_i - \hat{\beta}_i) = 0$ //

Q3

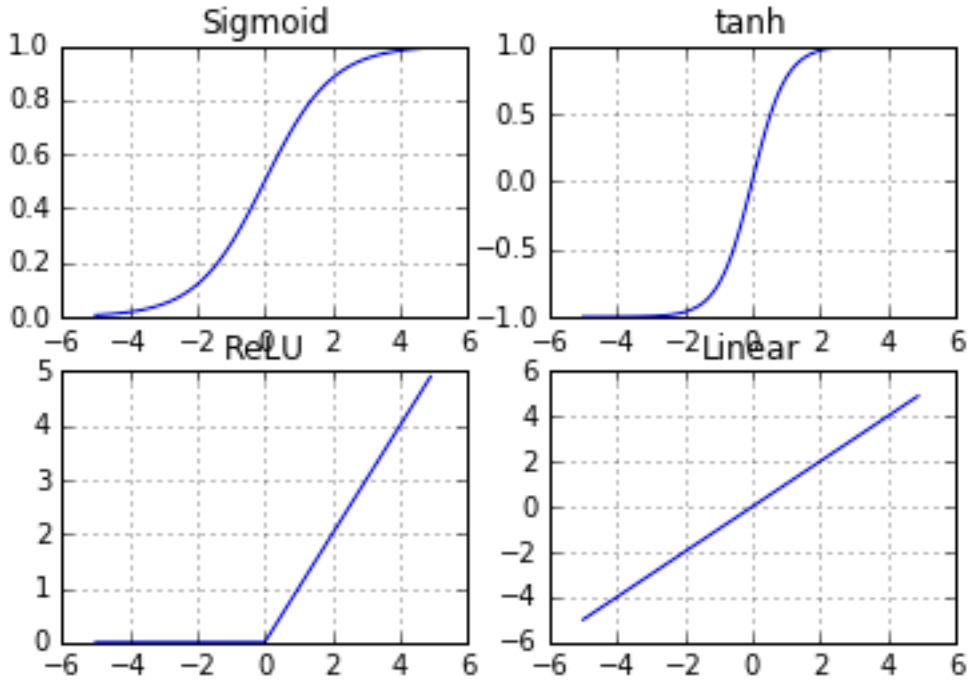
Optimal performance as measured by net loss is achieved with the following.
Learning rate = 0.3
Activation = sigmoid
Regularization = L2
Regularization rate = 0

Test Loss = 0.005
Training loss = 0.01



Learning rate controls the speed of gradient descent. Given a large learning rate, the loss function would fluctuate without converging. The learning rate is set reasonably slow – this doesn't matter with enough epochs.

In terms of **activation function**, tanh and sigmoid perform similarly well. ReLU does less well and linear is the worst. This is unsurprising, because tanh and sigmoid may as well be the same thing, up to different scaling.



Adding more **hidden layers** does not help at all. The best result is achieved with 1 hidden layer. The output form neurons of this hidden layer resemble to a large degree spirals or circles. Such outputs could not be produced with more layers.

Increasing **regularization rate** worsens Test Loss. This is surprising, probably because of the high 8:2 training and test data ratio. With 1:1 training and test data ratio, regularization improves performance by penalizing overfitting.

One could also note that there is an additional element of overfitting – model overfitting. This is the case where the experimenter chooses the best model & parameters for this particular set of test data, and the trained model performs badly on other sets of test data.

Noise have been added for more stable learning.

~~$\frac{\partial P_w(y|x)}{\partial w}$~~

$$\mathcal{L}(w) = \log \prod_{i=1}^{N} P_w(y_i|x_i)$$

$$= \sum_{i=1}^{N} \left(y_i w^T x_i - \log\left(1 + \exp(w^T x_i)\right) \right)$$

$$\nabla_w \mathcal{L}(w) = \sum_{i=1}^{N} \left(y_i x_i - \frac{\exp(w^T x_i)}{1 + \exp(w^T x_i)} x_i \right)$$

As in lecture slides, let $\mu_i = \frac{\exp(w^T x_i)}{1 + \exp(w^T x_i)}$

$$\nabla_w \mathcal{L}(w) = \sum_{i=1}^{N} (y_i - \mu_i) x_i \;//$$

$$\nabla_w^2 \mathcal{L}(w) = -\sum_{i=1}^{N} \nabla_w \mu_i \, x_i^T$$

$$H_{\mathcal{L}} = \nabla_w^2 \mathcal{L}(w) = -\sum_{i=1}^{N} \mu_i (1-\mu_i) x_i x_i^T$$

$$= -XRX^T$$

where $R_{ii} = \mu_i(1-\mu_i)$ //

$$\mu_i = \frac{\exp(w^T x_i)}{1 + \exp(w^T x_i)}$$

$$\nabla \mu_i = \frac{(1+e^{w^T x_i}) e^{w^T x_i} x_i - e^{w^T x_i} e^{w^T x_i} x_i}{\left(1 + \exp(w^T x_i)\right)^2}$$

$$= \frac{e^{w^T x_i} x_i}{\left(1 + e^{w^T x_i}\right)^2}$$

$$= \frac{1}{1+e^{w^T x_i}} \frac{e^{w^T x_i}}{1+e^{w^T x_i}} x_i$$

$$= \mu_i (1-\mu_i) x_i$$

Newton-Raphson:

$$x := x - \frac{f(x)}{f'(x)}$$

Let $x = w$ & $f(x) = \nabla_w \mathcal{L}(w)$

（科目：　　）**数　学　作　业　纸**

编号：　　班级：　　姓名：　　第　　页

IRLS:

$$w := w - H_L^{-1} \nabla_w L(w)$$
$$= w + (XRX^T)^{-1} X(y-\mu)$$
$$= (XRX^T)^{-1} (XRX^T w + X(y-\mu))$$
$$= (XRX^T)^{-1} XR [X^T w + R^{-1}(y-\mu)]$$ //

L2-norm regularised logistic regression

Let $-\frac{\lambda}{2}\|w\|_2^2 + L(w)$ be denoted by $\xi(w)$

$$\xi(w) = -\frac{\lambda}{2}\|w\|_2^2 + \sum_{i=1}^{N} (y_i w^T x_i - \log(1 + e^{w^T x_i}))$$

$$\nabla_w \xi(w) = -\lambda w + \nabla_w L(w)$$
$$= -\lambda w + X(y-\mu)$$ //

$$H_\xi = \nabla_w^2 \xi(w) = -\lambda + H_L$$
$$= -\lambda - XRX^T$$ //

IRLS:

$$w := w - H_\xi^{-1} \nabla_w \xi(w)$$
$$= w + (\lambda + XRX^T)^{-1} (-\lambda w + X(y-\mu))$$
$$= (\lambda + XRX^T)^{-1} (XRX^T w + X(y-\mu))$$
$$= (\lambda + XRX^T)^{-1} XR [X^T w + R^{-1}(y-\mu)]$$ //

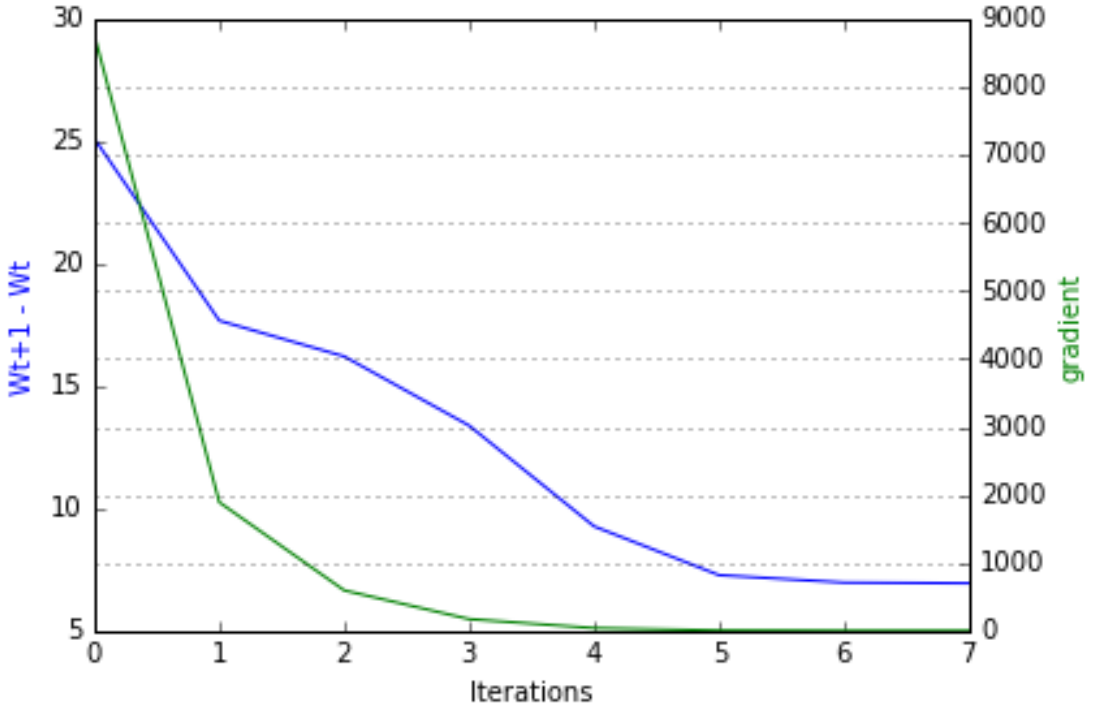
Q4

The main function *irls()* implements the IRLS algorithm according to the second last equation on the previous page.

Convergence

For both the regularized and unregularized versions, the trained weights demonstrate convergence, as measured by $\mathbf{w_{t+1} - w_t}$, and the gradient $\nabla\xi(\mathbf{w})$. For the unregularized version, $w_{t+1} - w_t$ stays at ~7 and $\nabla\xi(w)$ below 0.1. These values are thus set as tolerance levels to stop *irls()*.

Fig 1: Convergence of unregularized IRLS algorithm



Regularization

Next, we test a few candidate values for λ, namely [0.001, 0.01, 0.1, 1, 10]. The training set is split into a **regularized training set** (66%, 21490 samples) and a **cross validation set** (34%, 11071 samples). For each λ, the weights are learned with the regularized training set. The λ with the best prediction accuracy on the cross validation set is then selected, and turns out to be 0.1.

Performance

We compare prediction on test data with λ=0 and λ=0.1. Results below.

Table 1: Performance metrics of different λ values

λ	Number of iterations	Training accuracy	Cross validation accuracy	Test accuracy	L2 Norm
0/unregularize	7	84.91%	-	85.00%	15.12
0.001	8	84.91%	84.93%	-	-
0.01	8	84.91%	84.93%	-	-
0.1	7	84.91%	84.95%	84.99%	7.94
1	7	84.85%	84.89%	-	-
10	7	84.77%	84.78%	-	-

Choosing a large value for λ is apparently a bad idea. But besides, there is not much difference between the unregularized and regularized versions. The regularized ISRL similarly converges, albeit to a tighter bound.

Fig 2: Convergence of regularized IRLS algorithm where λ=0.1