Machine Learning - Homework 2

資工四 B05902023 李澤諺

March 26, 2020

1. (2%) 請比較實作的 generative model 及 logistic regression 的準確率，何者較佳? 請解釋爲何有這種情況。

我選出了 1000 筆 data 作爲 validation data，剩下的則作爲 training data，而我實作中 performance 最好的 generative model 爲：先將所有 data 之中 age 的值提升至 2 到 30 次方，並將所有 feature 進行 normalization 後，再建立 generative model，而我實作中 performance 最好的 logistic regression 爲：先將所有 data 之中 age 的值提升至 2 到 10 次方，並將所有 feature 進行 normalization，接著，我實作了 Adagrad 進行 logistic regression，其中，我使用了 batch gradient descent，learning rate 爲 0.05，並且使用了 L2 regularization，其中 regularization 的權重爲 0.001，訓練了 1000 個 epoch，以此得到最後的 model。下表爲以上所述的 generative model 和 logistic regression 分別在 training 和 testing 時所得到的 accuracy：

	Train	Validation	Test	
			Public	Private
Generative Model	0.87004	0.87000	0.87763	0.87669
Logistic Regression	0.88576	0.89200	0.89023	0.89124

由上表可以看出我實作的 model 中，logistic regression 的 performance 較 generative model 好，我認爲其原因爲：generative model 對 data 的 distribution 進行假設，然而其假設可能不甚正確，使得其 performance 較未對 data 的 distribution 進行假設的 logistic regression 還要來得差。

2. (2%) 請實作 logistic regression 的正規化 (regularization)，並討論其對於你的模型準確率的影響。接著嘗試對正規項使用不同的權重 (lambda)，並討論其影響。(有關 regularization 請參考https://goo.gl/SSWGhf p.35)

我選出了 1000 筆 data 作爲 validation data，剩下的則作爲 training data，而我實作的 logistic regression 爲：先將所有 data 之中 age 的值提升至 2 到 10 次方，並將所有 feature 進行 normalization，接著，我實作了 Adagrad 進行 logistic regression，其中，我使用了 batch gradient descent，learning rate 爲 0.05，並且使用了 L2 regularization，其中 regularization 的權重分別爲 0 (即沒有 regularization)、0.001、0.01、0.1，訓練了 1000 個 epoch，以此得到各個 model。下表爲以上所述的 logistic regression 分別在 training 和 testing 時所得到的 accuracy：

	Train	Validation	Test	
			Public	Private
$\lambda = 0$	0.88503	0.89000	0.89023	0.89102
$\lambda = 0.001$	0.88576	0.89200	0.89023	0.89124
$\lambda = 0.01$	0.88570	0.88800	0.89008	0.89088
$\lambda = 0.1$	0.88505	0.88700	0.89001	0.89001

由上表可以看出，當 regularization 的權重爲 0.001 時，其 performance 比沒有使用 regularization 時還要再高了一些，其可能是因爲在沒有進行 regularization 的情況下，model 的複雜度太高，其與最好的 model 有一些差距，會使得其 performance 較差，因此在加入了一些 regularization 去限制 model 的複雜度之後，可以使得 model 的 performance 上升一些，而當 regularization 的權重爲 0.01 與 0.1 時，其 performance 比沒有使用 regularization 時還要再差了一些，其可能是因爲 regularization 的權重太高，即對 model 複雜度的限制太多，使得其與最佳的 model 相去甚遠所致。

3. (1%) 請說明你實作的 best model，其訓練方式和準確率爲何?

我選出了 1000 筆 data 作爲 validation data，剩下的則作爲 training data，而我實作中 performance 最好的 model 爲：先將所有 data 之中 age 的值提升至 2 到 10 次方，並將所有 feature 進行 normalization，接著，我實作了 Adagrad 進行 logistic regression，其中，我使用了 batch gradient descent，learning rate 爲 0.05，並且使用了 L2 regularization，其中 regularization 的權重爲 0.001，訓練了 1000 個 epoch，以此得到最後的 model。下表爲以上所述的 logistic regression 分別在 training 和 testing 時所得到的 accuracy：

	Train	Validation	Test	
			Public	Private
Logistic Regression	0.88576	0.89200	0.89023	0.89124

4. (1%) 請實作輸入特徵標準化 (feature normalization)，並比較是否應用此技巧，會對於你的模型有何影響。

下表爲我實作的 generative model (方法如第 1 題所述)，有使用 normalization 和沒有使用 normalization 分別在 training 和 testing 時所得到的 accuracy：

	Train	Validation	Test	
			Public	Private
Normalization	0.87004	0.87000	0.87763	0.87669
Without Normalization	0.79403	0.81800	0.80348	0.79472

下表爲我實作的 logistic regression (方法如第 1 題所述)，有使用 normalization 和沒有使用 normalization 分別在 training 和 testing 時所得到的 accuracy：

	Train	Validation	Test	
			Public	Private
Normalization	0.88576	0.89200	0.89023	0.89124
Without Normalization	0.74683	0.76000	0.58388	0.58931

事實上，在沒有使用 normalization 時，計算上很容易發生 overflow，並且，沒有使用 normalization 會使得每個 feature 的重要程度不一，由上表可以看出，有使用 normalization 時的 performance 會比沒有使用 normalization 還要好上許多。