ECGR 5105 Homework 2

Owen Bailey-Waltz (801488178)
GitHub Link: https://github.com/obaileyw-uncc/ecgr5105/hw02_paramscaling_regularization

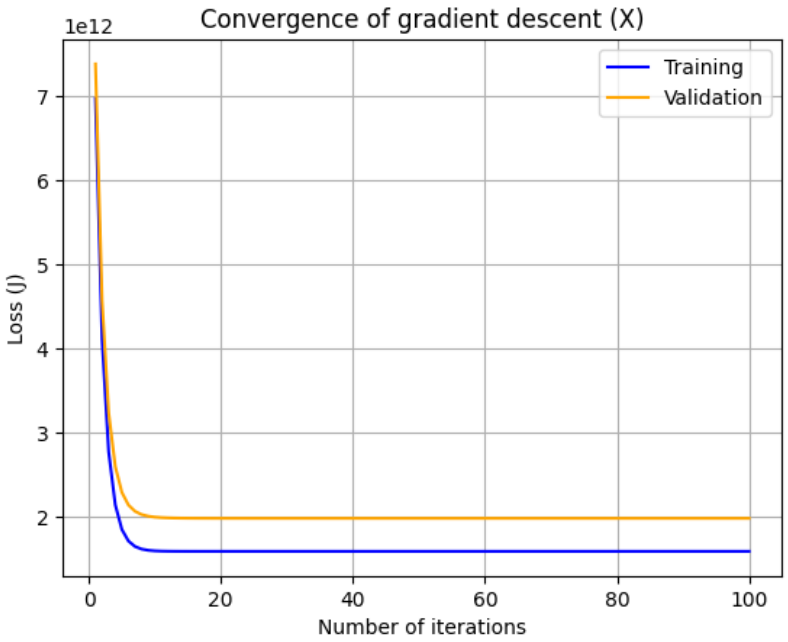
Problem 1: Models with no input scaling

(a) Multi-variable gradient descent: area, bedrooms, bathrooms, stories, parking

Values of α between 0.1 and 0.01 cause gradient descent on the unadjusted input data to diverge. For a convergent response, a learning rate of approximately 10^{-8} is required. With a learning rate of 10^{-8}, the convergence of the unadjusted model is poor as the model is overfitted, with a large gap in training and validation loss. The convergent response with $\mu = 10^{-8}$ is shown.

$$h(\boldsymbol{X}) = 0.47971 + 852.65x_1 + 1.7381x_2 + 0.90963x_3 + 1.3630x_4 + 0.33500x_5$$

$$J(\boldsymbol{\Theta}) = 1.595 \times 10^{12} \qquad J_{\text{val}}(\boldsymbol{\Theta}) = 1.986 \times 10^{12}$$

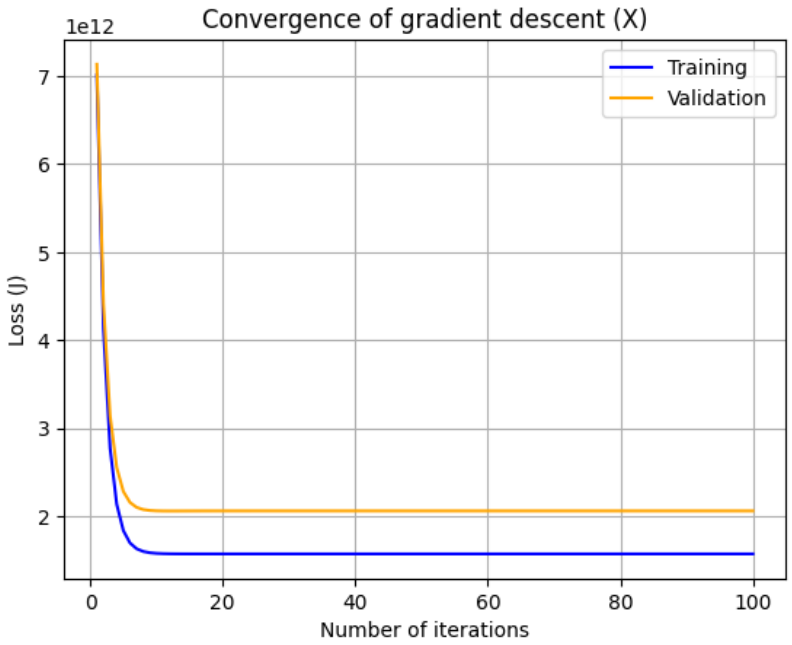


(b) Multi-variable gradient descent: all variables but "furnished"

Values of α between 0.1 and 0.01 cause gradient descent on the unadjusted input data to diverge. For a convergent response, a learning rate of approximately 10^{-8} is required. With a learning rate of 10^{-8}, the convergence of the unadjusted model is poor as the model is overfitted, with a large gap in training and validation loss. Compared to the model with fewer parameters, the model converges to one with a slightly lower training loss and a slightly higher validation loss. The convergent response with $\mu = 10^{-8}$ is shown.

$$h(\boldsymbol{X}) = 0.49875 + 857.38x_1 + 1.7532x_2 + 0.90658x_3 + 1.4521x_4 + 0.43710x_5 + 0.17889x_6 + 0.32816x_7 + 0.055977x_8 + 0.34316x_9 + 0.35553x_{10} + 0.19163x_{11}$$

$$J(\boldsymbol{\Theta}) = 1.575 \times 10^{12} \qquad J_{\text{val}}(\boldsymbol{\Theta}) = 2.063 \times 10^{12}$$



Problem 2: Input-scaled models

(a) Problem 1(a) with normalization and standardization

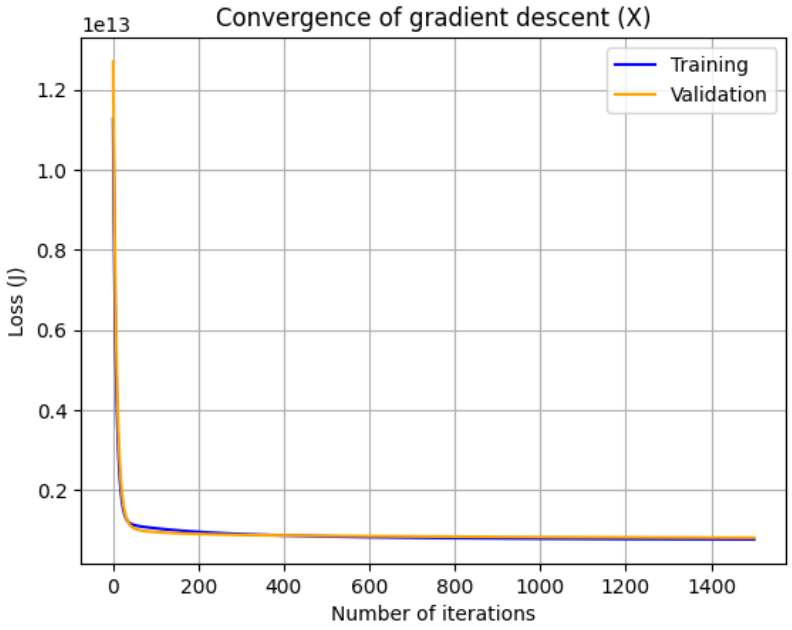
A learning rate over 1500 iterations and a learning rate of 0.05 was used for this problem as learning rates between 0.01 and 0.1 allow the model to converge. Both the model trained with minimum-maximum normalized input features and Gaussian standardized input features produced significantly lower loss as well as differences in validation and training loss than the baseline training in Problem 1, with standardization producing a lower actual

loss and a difference in validation loss that is smaller by a full order of magnitude (-4.37×10^9 compared to 4.07×10^{10}), making standardization more effective than min-max normalization for this model.

Normalization

$$h(\boldsymbol{X}) = 2198448 + 3780249x_1 + 1283686x_2 + 3185855x_3 + 1738521x_4 + 1490609x_5$$

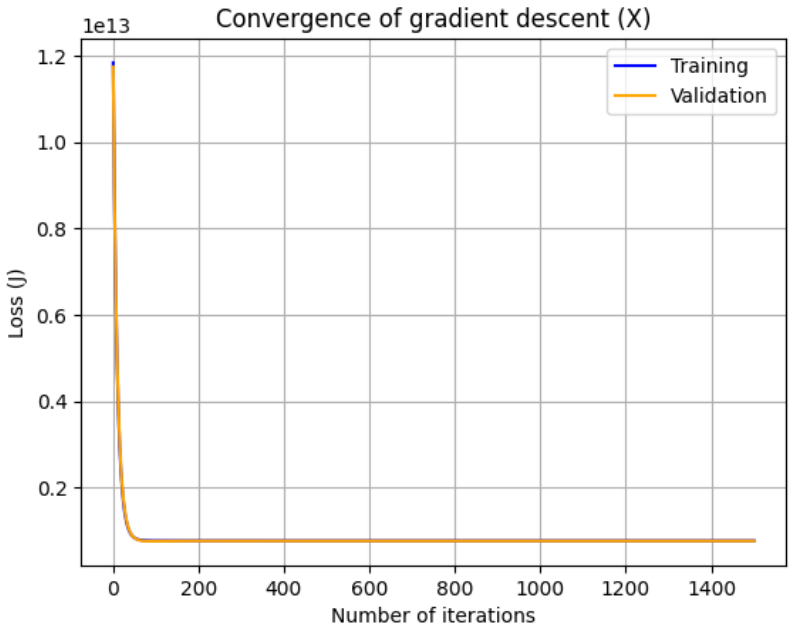
$$J(\boldsymbol{\Theta}) = 7.720 \times 10^{11} \qquad J_{\text{val}}(\boldsymbol{\Theta}) = 8.128 \times 10^{11}$$



Standardization

$$h(\boldsymbol{X}) = 4747684 + 749416x_1 + 71898x_2 + 628129x_3 + 467497x_4 + 293627x_5$$

$$J(\boldsymbol{\Theta}) = 7.692 \times 10^{11} \qquad J_{\text{val}}(\boldsymbol{\Theta}) = 7.648 \times 10^{11}$$



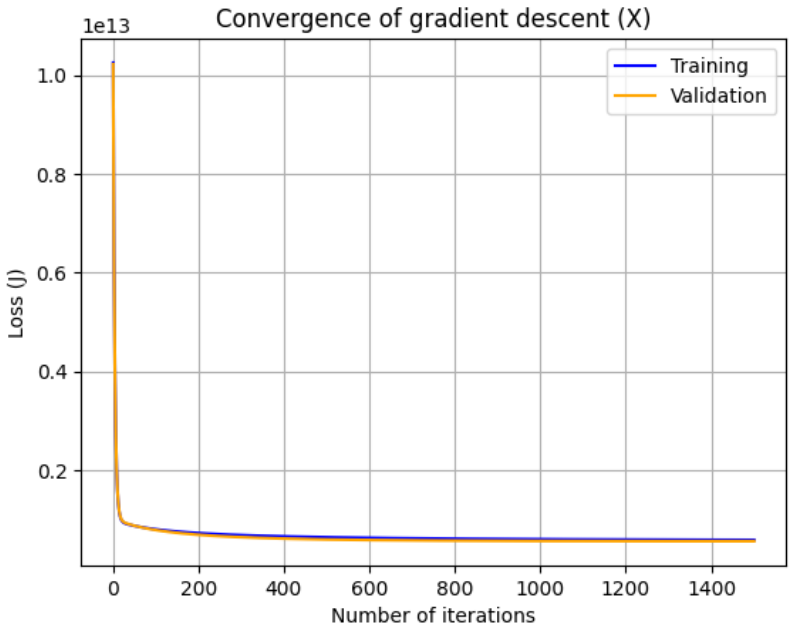
(b) Problem 1(b) with normalization and standardization

A learning rate over 1500 iterations and a learning rate of 0.05 was used for this problem as learning rates between 0.01 and 0.1 allow the model to converge. Both the model trained with minimum-maximum normalized input features and Gaussian standardized input features produced significantly lower loss as well as differences in validation and training loss than the baseline training in Problem 1. For this problem, normalization produced a lower difference between training and validation loss and lower validation loss level, while standardization produced a lower training loss with a higher validation loss. Since the goal of a model is to generalize, the lower validation loss of minimum-maximum normalization makes that form of feature scaling work better on this more-detailed model, which may be due to the presence of the binary feature inputs that were not part of the less-detailed model.

Normalization

$$h(\boldsymbol{X}) = 1777166 + 2815335x_1 + 1016315x_2 + 2749871x_3 + 1245824x_4 + 509103x_5 + 363726x_6 + 386192x_7 + 861773x_8 + 894166x_9 + 833299x_{10} + 693202x_{11}$$

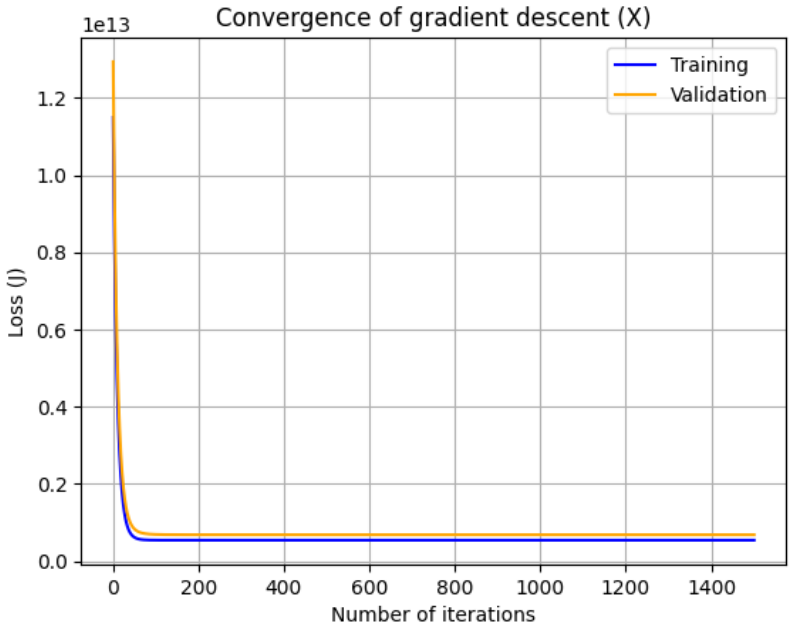
$$J(\boldsymbol{\Theta}) = 5.844 \times 10^{11} \qquad J_{\text{val}}(\boldsymbol{\Theta}) = 5.583 \times 10^{11}$$



Standardization

$$h(\boldsymbol{X}) = 4749014 + 534348x_1 + 97970x_2 + 431243x_3 + 428637x_4 + 130230x_5 + 162189x_6 + 177745x_7 + 206256x_8 + 419006x_9 + 266853x_{10} + 282602x_{11}$$

$$J(\boldsymbol{\Theta}) = 5.459 \times 10^{11} \qquad J_{\text{val}}(\boldsymbol{\Theta}) = 6.871 \times 10^{11}$$



Problem 3: Input-scaled models with parameter penalties applied

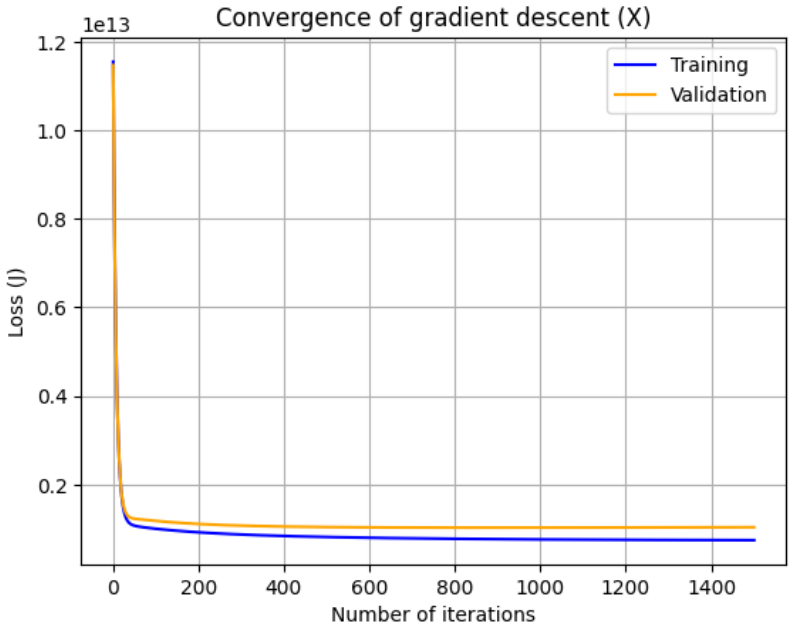
(a) Problem 2(a) with normalization and standardization

A learning rate over 1500 iterations, a learning rate of 0.05, and a λ value of 0.5 was used for this problem. Introducing parameter penalties caused the validation loss to increase compared to the training in Problem 2 while the training loss decreased for both forms of pre-processing. Normalization and standardization produced similar differences between training and validation losses. For this model, standardization has a lower validation loss and is more effective.

Normalization

$$h(\boldsymbol{X}) = 2311551 + 4033087x_1 + 1250736x_2 + 2934648x_3 + 1672624x_4 + 1228454x_5$$

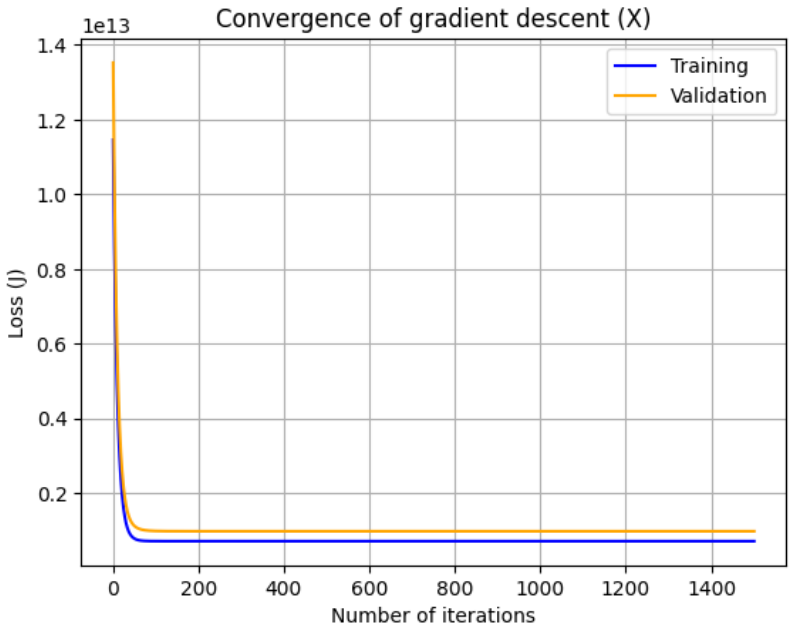
$$J(\boldsymbol{\Theta}) = 7.451 \times 10^{11} \qquad J_{\text{val}}(\boldsymbol{\Theta}) = 9.670 \times 10^{11}$$



Standardization

$$h(\boldsymbol{X}) = 4728120 + 721160x_1 + 130807x_2 + 553500x_3 + 464587x_4 + 272734x_5$$

$$J(\boldsymbol{\Theta}) = 7.163 \times 10^{11} \qquad J_{\text{val}}(\boldsymbol{\Theta}) = 9.765 \times 10^{11}$$



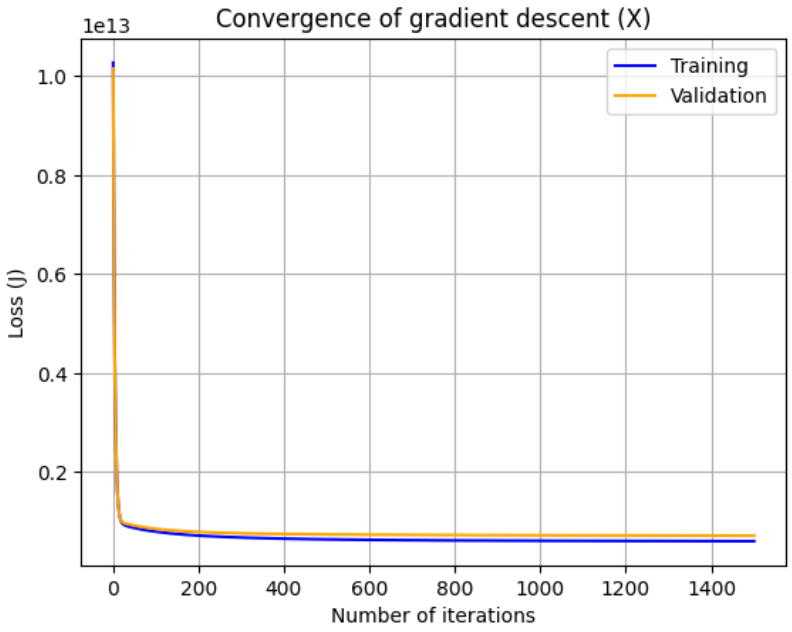
(b) Problem 2(b) with normalization and standardization

A learning rate over 1500 iterations, a learning rate of 0.05, and a λ value of 1 was used for this problem. Introducing parameter penalties caused the validation loss to increase for normalization while the training loss remained approximately the same. Meanwhile, training loss increased slightly and validation loss decreased for the standardized case. Standardization is clearly the more effective form of input scaling for this model as both loss levels are considerably lower than with normalization.

Normalization

$$h(\boldsymbol{X}) = 1686841 + 2612544x_1 + 1244947x_2 + 2368014x_3 + 1341966x_4 + 519554x_5 + 303412x_6 + 375954x_7 + 920961x_8 + 960961x_9 + 1076982x_{10} + 710118x_{11}$$

$$J(\boldsymbol{\Theta}) = 6.000 \times 10^{11} \qquad J_{\text{val}}(\boldsymbol{\Theta}) = 6.227 \times 10^{11}$$



Standardization

$$h(\boldsymbol{X}) = 4777430 + 598878x_1 + 95587x_2 + 473122x_3 + 355994x_4 + 164978x_5 + 114089x_6 + 152363x_7 + 159392x_8 + 406160x_9 + 251785x_{10} + 253773x_{11}$$

$$J(\boldsymbol{\Theta}) = 5.736 \times 10^{11} \qquad J_{\text{val}}(\boldsymbol{\Theta}) = 5.843 \times 10^{11}$$