Data Science Methods for Finance

Freek Linssen

Predicting Stock Returns with Momentum

Question 1.1, 1.2 & 1.3

	Return	Mom_1	Mom_2-1	Mom_3-1	Mom_4-1	Mom_5-1	Mom_6-1	Mom_7-1	Mom_8-1	Mom_9-1	Mom_10-1	Mom_11-1	Mom_12-1
1.1													
Mean Return	0.95%	0.97%	0.99%	1.92%	2.90%	3.92%	4.95%	5.97%	7.03%	8.16%	9.35%	10.55%	11.80%
Missing Values	0	14,307	28,475	42,518	46,408	70,186	83,806	97,209	110,492	123,650	136,690	149,597	162,378
1.2													
Mean Return	0.95%	0.97%	0.98%	1.91%	2.86%	3.85%	4.83%	5.81%	6.82%	7.89%	8.99%	10.10%	11.25%
Missing Values	0	4,728	9,424	14,028	18.706	23.287	27.827	32.285	36.705	41.086	45.442	49.762	54.052
1.3													
Mean Return	0.95%	0.96%	0.97%	1.89%	2.82%	3.79%	4.75%	5.69%	6.67%	7.69%	8.73%	9.79%	10.87%
Missing Values	0	0	0	0	0	0	0	0	0	0	0	0	0

Question 1.4

	Mom_1	Mom_2-1	Mom_3-1	Mom_4-1	Mom_5-1	Mom_6-1	Mom_7-1	Mom_8-1	Mom_9-1	Mom_10-1	Mom_11-1	Mom_12-1
Mean Return	0.19	0.17	-0.19	-0.33	-0.41	-0.46	-0.51	-0.55	-0.57	-0.60	-0.62	-0.64
SD	0.27	0.27	0.24	0.24	0.23	0.23	0.23	0.23	0.23	0.23	0.23	0.23

Question 1.5

	1990-01-01 - 1999-12-31	2000-01-01 - 2007-06-30	2007-07-01 - 2021-12-31	2010-01-01 - 2021-12-31
Mean Return	1.10%	1.17%	0.64%	1.00%
Standard Deviation	16.0%	17.5%	14.6%	13.1%

Question 1.6

	ret	mom1	mom_2_1	mom_3_1	mom_4_1	mom_5_1	mom_6_1	mom_7_1	mom_8_1	mom_9_1	mom_10_1	mom_11_1	mom_12_1
ret	1.00	-0.04	0.01	0.04	0.03	0.04	0.03	0.04	0.03	0.03	0.05	0.04	0.04
mom1	-0.04	1.00	0.03	0.09	0.10	0.09	0.09	0.11	0.10	0.13	0.11	0.10	0.10
mom_2_1	0.01	0.03	1.00	0.67	0.53	0.45	0.40	0.36	0.34	0.31	0.31	0.28	0.26
mom_3_1	0.04	0.09	0.67	1.00	0.78	0.68	0.61	0.57	0.52	0.47	0.47	0.45	0.43
mom_4_1	0.03	0.10	0.53	0.78	1.00	0.82	0.73	0.66	0.58	0.53	0.52	0.52	0.49
mom_5_1	0.04	0.09	0.45	0.68	0.82	1.00	0.85	0.79	0.70	0.64	0.63	0.59	0.58
mom_6_1	0.03	0.09	0.40	0.61	0.73	0.85	1.00	0.86	0.78	0.72	0.69	0.67	0.64
mom_7_1	0.04	0.11	0.36	0.57	0.66	0.79	0.86	1.00	0.87	0.80	0.77	0.72	0.71
mom_8_1	0.03	0.10	0.34	0.52	0.58	0.70	0.78	0.87	1.00	0.87	0.83	0.77	0.74
mom_9_1	0.03	0.13	0.31	0.47	0.53	0.64	0.72	0.80	0.87	1.00	0.89	0.84	0.79
mom_10_1	0.05	0.11	0.31	0.47	0.52	0.63	0.69	0.77	0.83	0.89	1.00	0.90	0.85
mom_11_1	0.04	0.10	0.28	0.45	0.52	0.59	0.67	0.72	0.77	0.84	0.90	1.00	0.91
mom_12_1	0.04	0.10	0.26	0.43	0.49	0.58	0.64	0.71	0.74	0.79	0.85	0.91	1.00

1
0.5
0
-0.5
-1

The correlation matrix shows an overview of the correlation between the return and the momentum variables and the correlation between the momentum variables with each other.
What stands out is the low correlation between the return and the momentum variables, this can be worrisome because the low correlation indicates a low predictive power of the momentum variables.
The negative correlation between return and mom1 is also noticeable since this confirms the short-term reversal effect.

Question 2.1

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.0464571  0.0010137  45.827  < 2e-16 ***
mom1        -0.0318103  0.0008342 -38.132  < 2e-16 ***
mom_2_1     -0.0180533  0.0011088 -16.282  < 2e-16 ***
mom_3_1      0.0272127  0.0017540  15.515  < 2e-16 ***
mom_4_1      0.0009502  0.0020461   0.464  0.64236    
mom_5_1      0.0049295  0.0023952   2.058  0.03959 *  
mom_6_1      0.0016365  0.0024913   0.657  0.51125    
mom_7_1      0.0079264  0.0027768   2.854  0.00431 ** 
mom_8_1     -0.0199889  0.0028602  -6.989 2.78e-12 ***
mom_9_1     -0.0268601  0.0030606  -8.776  < 2e-16 ***
mom_10_1     0.0959463  0.0034621  27.713  < 2e-16 ***
mom_11_1    -0.0259599  0.0036552  -7.102 1.23e-12 ***
mom_12_1    -0.0034433  0.0030704  -1.121  0.26210    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1597 on 591178 degrees of freedom
Multiple R-squared:  0.005891,  Adjusted R-squared:  0.005871 
F-statistic: 291.9 on 12 and 591178 DF,  p-value: < 2.2e-16
```

Kelly Gu R-squared, training set: 0.010

Kelly Gu R-squared, validation set: 0.004

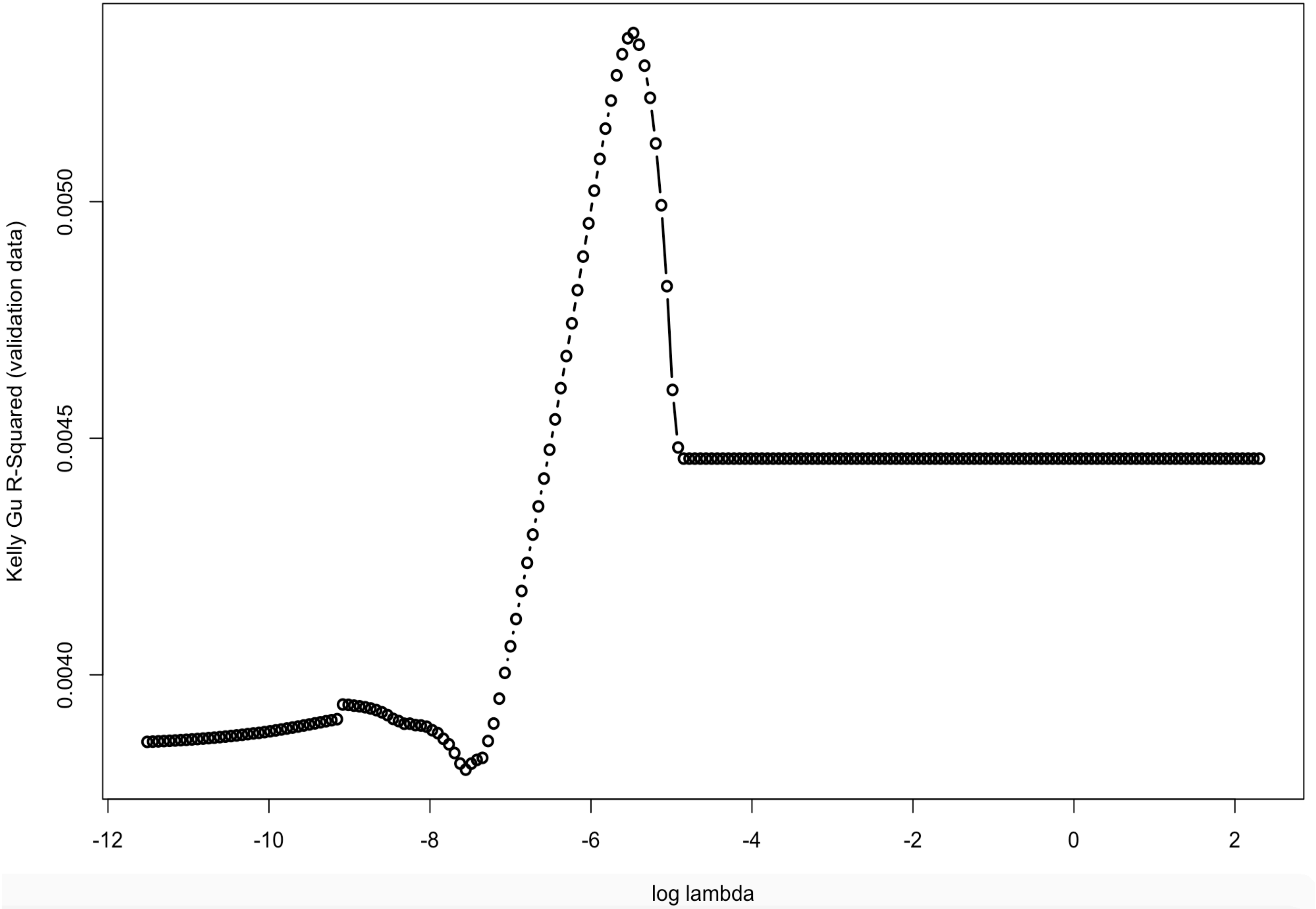
Kelly Gu R-squared, test set including crisis: -0.005

Kelly Gu R-squared, test set excluding crisis: 0.002

The regression output shows clearly that momentum has a significant effect on returns since eight of the momentum variables have a highly significant t-value above three.

Looking at the Kelly Gu R-squared values, we can observe that the predictive power of this model is very low in the test samples compared to the training sample.
For example; the Kelly Gu R-squared is negative in the test set including the crisis, this means that a straight line is a better predictor than this linear regression. The relatively low predictive power outside the training sample indicates overfitting. So, other models that prevent overfitting might be better in this case.

Question 2.2



The highest Kelly Gu R-squared I found in the validation set was 0.0054.
This value was reached with a Lambda of 0.004199.

This optimal Lambda was found with a Lambda grit between 0.00001 and 1 of length 200.

Question 2.3

```
Optimal Lasso Regression
13 x 1 sparse Matrix of class "dgCMatrix"
                       s0
(Intercept)  0.02667572
mom1        -0.01253121
mom_2_1      .
mom_3_1      .
mom_4_1      .
mom_5_1      .
mom_6_1      .
mom_7_1      .
mom_8_1      .
mom_9_1      .
mom_10_1     0.02108811
mom_11_1     .
mom_12_1     .
```

We can observe that in this optimal Lasso model 10 of the 12 coefficients of the momentum variables are set to zero, this is caused by the shrinkage parameter in a Lasso model.
Setting the coefficients from less important variables to zero prevent a model from overfitting.

* It was not possible the incorporate significance levels with a glmnet model

Question 2.4

The Final Lasso Model:

Kelly Gu R-squared, test set including crisis: 0.0007

Kelly Gu R-squared, test set excluding crisis: 0.0055

Linear Model from 2.1:

Kelly Gu R-squared, test set including crisis: -0.005

Kelly Gu R-squared, test set excluding crisis: 0.002

We can clearly observe that the final Lasso model has a higher Kelly Gu R-squared in both test sets, this is the result of the shrinkage penalty in the Lasso model that prevents overfitting.

The better performance of the Lasso model compared to a linear model in the training sets clearly shows that the linear model was overfitting and that a shrinkage penalty is beneficial for this prediction task.

Question 3.1

Test set including crisis:

Average monthly return for the long portfolio : 0.98%

Average monthly return for the short portfolio: -0.06%

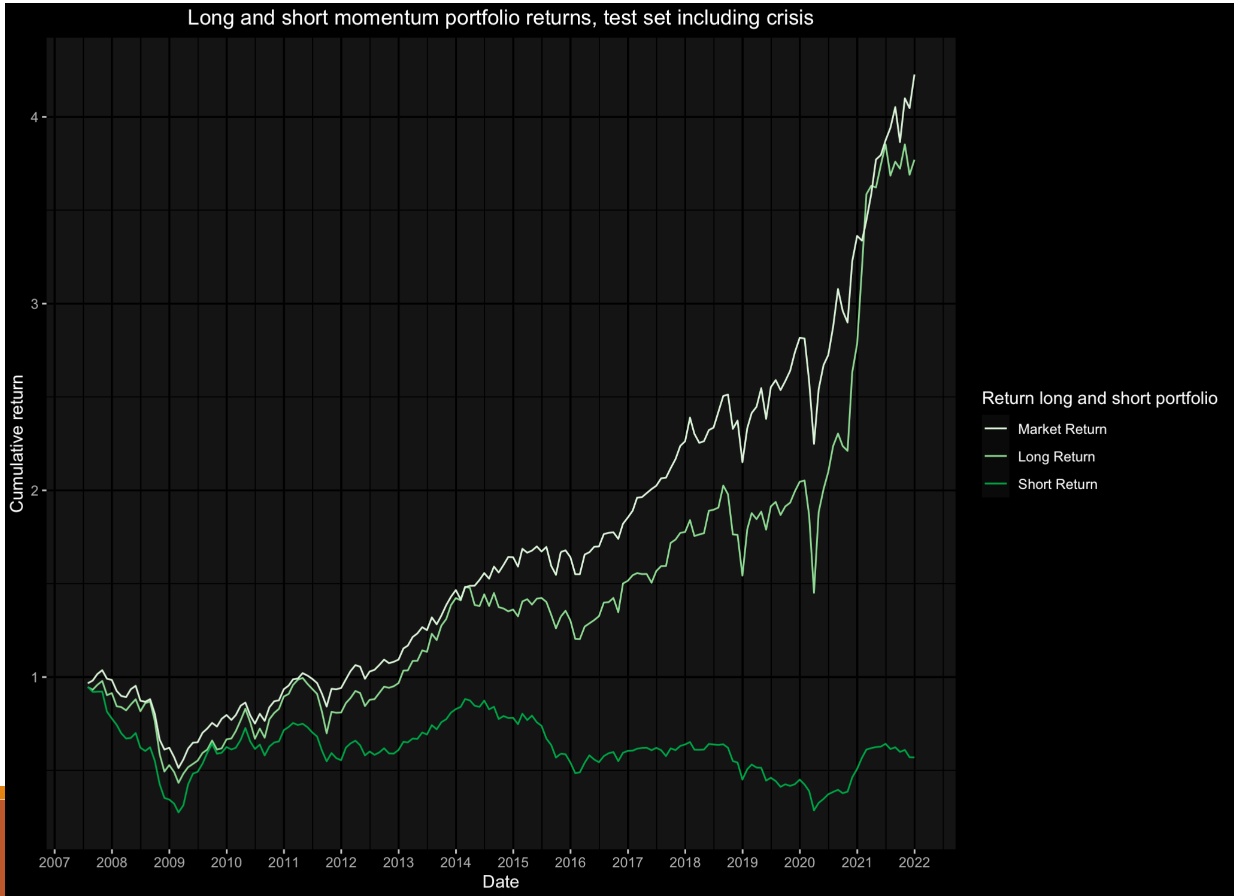
Average monthly value-weighted market return (reference): 0.94%

Test set excluding crisis:

Average monthly return for the long portfolio : 1.40%

Average monthly return for the short portfolio: 0.13%

Average monthly value-weighted market return (reference): 1.25%



Question 3.2

Test set including crisis:

Average monthly return for the momentum factor portfolio : 1.04%
Monthly standard deviation for the momentum factor portfolio: 3.86%

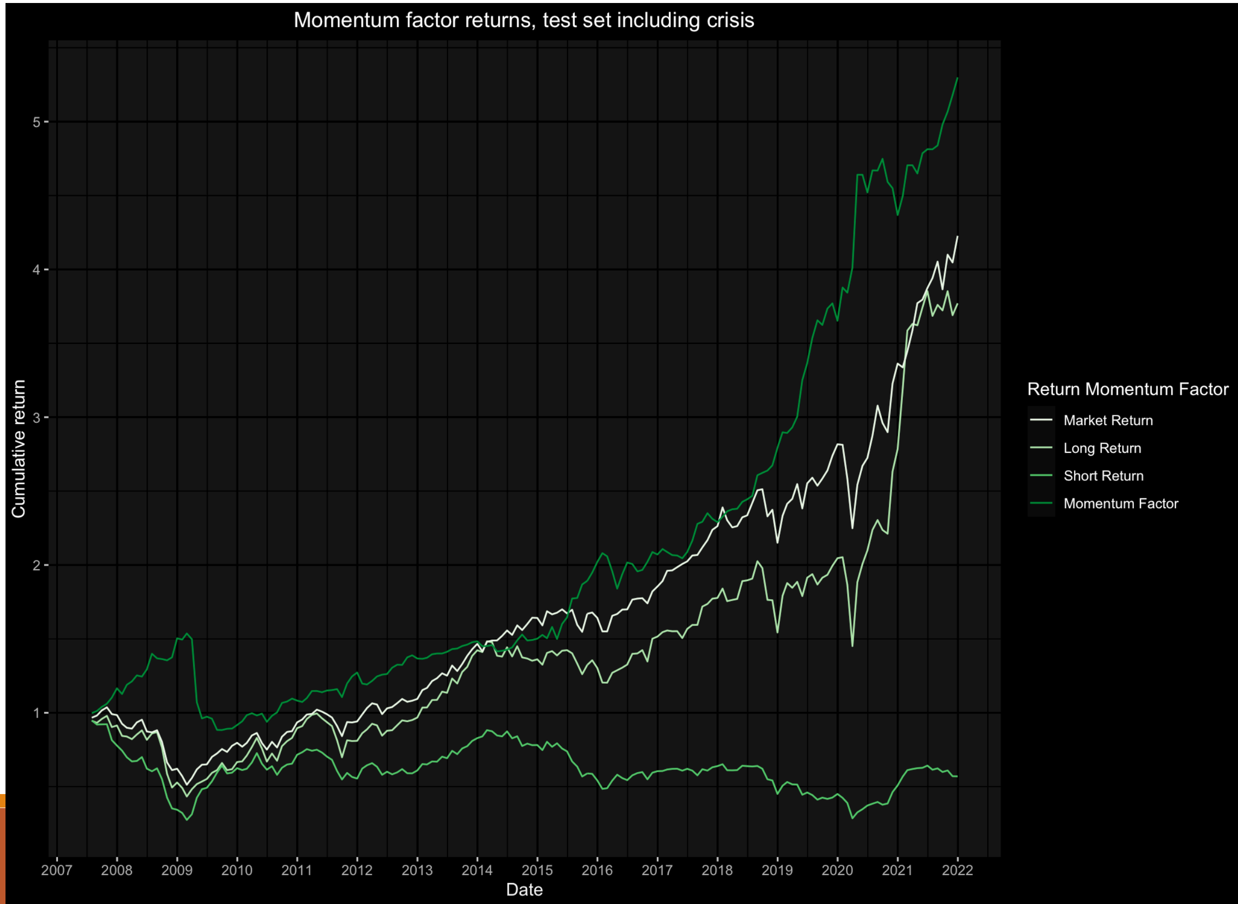
Average monthly value-weighted market return (reference): 0.94%
Monthly standard deviation value-weighted market return (reference): 4.57%

Test set excluding crisis:

Average monthly return for the momentum factor portfolio : 1.27%
Monthly standard deviation for the momentum factor portfolio: 2.94%

Average monthly value-weighted market return (reference): 1.25%
Monthly standard deviation value-weighted market return (reference): 4.09%

The average monthly return of the momentum factor is higher than the average monthly market return while having a lower standard deviation in both test datasets.
In the figure, we can also clearly see that the short portfolio is performing way worse than the market and that the return on the long portfolio is similar to the market return in the test set including the financial crisis.
This is an indication that the momentum strategy is working and that a Lasso model is useful for predicting future returns with momentum data.



Question 3.3

CAPM regression test set including crisis

	Dependent variable:
	Momentum_Factor
market_return	-0.071
	(0.064)
Constant	0.011^{***}
	(0.003)
Observations	174
R^2	0.007
Adjusted R^2	0.001
Residual Std. Error	0.039 (df = 172)
F Statistic	1.212 (df = 1; 172)
Note:	*p<0.1; **p<0.05; ***p<0.01

The CAPM regressions in both test sets show a highly significant alpha (outperformance) of the momentum factor relative to the value-weighted market portfolio.

This highly positive and significant constant clearly shows that the momentum strategy (going long in stocks with a high momentum and going short in stocks with a low momentum) is working and that a Lasso model is helpful for assigning stocks to the long and short portfolio.

CAPM regression test set excluding crisis

	Dependent variable:
	Momentum_Factor
market_return	0.052
	(0.060)
Constant	0.012^{***}
	(0.003)
Observations	144
R^2	0.005
Adjusted R^2	-0.002
Residual Std. Error	0.029 (df = 142)
F Statistic	0.744 (df = 1; 142)
Note:	*p<0.1; **p<0.05; ***p<0.01

Question 4.1

Network Architecture:

```
model <- keras_model_sequential() %>%
layer_dense(units = 4, activation = "relu", input_shape = dim(x_train)[2]) %>%
layer_dense(units = 1, activation = "linear")
```

I chose a linear activation function in the last layer and tthe mean squared error as loss function since the target values are continuous and not categorical.

Advantages and disadvantages of Neural Networks:

The main advantage of neural networks is that they are very flexible and therefore able to detect highly non-linear structures in the data. Neural networks can be fitted to many types of data: time-series, cross-sectional, photos, videos, and audio.

Another advantage is that a neural network finds the optimal parameters itself by backpropagation, so you don’t have to find any parameters like Lambda yourself. This leads however to one of the main disadvantages for neural networks: they tend to overfit very easily, there are however various regularization methods to prevent this.

One other disadvantage is that training neural networks takes a lot of computation, especially if the model has many neurons and multiple layers, this results in a long training time and high energy usage.

Question 4.2

Mean Squared Error, training set (last epoch): 0,0256

Accuracy, training set (last epoch): 0.0523

Kelly Gu R-squared, training set: 0.0101

Kelly Gu R-squared, validation set: 0.0025 (Lasso: 0.0054)

Kelly Gu R-squared, test set including crisis: -0.0087 (Lasso: 0.0007)

Kelly Gu R-squared, test set excluding crisis: -0.0021 (Lasso: 0.0055)

The Kelly Gu R-squared values show that this simple neural network with one hidden layer and without regularization is performing worse on the validation and test sets than the Lasso model.

Especially the negative Kelly Gu R-squared in both test sets stand out, these negative values indicate that a straight line is a better predictor than this neural network. These low values in both test sets in combination with the high Kelly Gu R-squared on the training set (higher than with the Lasso model) show that this neural network is overfitting. Hence, regularization methods should be added to prevent this model from overfitting.

Question 4.3

I also tried the same model with a drop-out layer (0.2) to prevent the network even more from overfitting, this improved the Kelly Gu R-squared on the validation and test sets even more.

Mean Squared Error, training set (last epoch): 0,0256	with drop-out: 0,0256	
Accuracy, training set (last epoch): 0.0523	with drop-out: 0.0523	
Kelly Gu R-squared, training set: 0.0098	with drop-out: 0.0089	
Kelly Gu R-squared, validation set: 0.0044	with drop-out: 0.0056	(Lasso: 0.0054)
Kelly Gu R-squared, test set including crisis: -0.0048	with drop-out: -0.0004	(Lasso: 0.007)
Kelly Gu R-squared, test set excluding crisis: 0.0011	with drop-out: 0.0053	(Lasso: 0.0055)

This neural network is performing way better on the validation and test sets compared to the neural network in question 4.3, this means that the kernel regularization (and drop-out) is successfully preventing the model from overfitting. The validation/test set performance of the neural network including regularization and drop-out is similar to the performance of the optimal Lasso model in 2.2.

I will use the model with drop-out to make the final predictions for 4.4 since this model has a higher Kelly Gu R-squared on the validation and test sets.

Question 4.4

Momentum Factor with the Final Lasso Model:

Test set including crisis: average return: 1.04%
standard deviation: 3.86%

Test set excluding crisis: average return: 1.27%
standard deviation: 2.94%

Momentum Factor with the Deep Learning Model including weight regularization and drop-out:

Test set including crisis: average return: 0.81%
standard deviation: 2.86%

Test set excluding crisis: average return: 0.81%
standard deviation: 2.38%

Value-Weighted Market Return (reference):

Test set including crisis: average return: 0.94%
standard deviation: 4.57%

Test set excluding crisis: average return: 1.25%
standard deviation: 4.09%

Deep Learning CAPM regression test set including crisis

	Dependent variable:
	factor_DL
market_return	-0.008
	(0.048)
Constant	0.008***
	(0.002)
Observations	174
R^2	0.0002
Adjusted R^2	-0.006
Residual Std. Error	0.029 (df = 172)
F Statistic	0.028 (df = 1; 172)
Note:	*p<0.1; **p<0.05; ***p<0.01

Deep Learning CAPM regression test set excluding crisis

	Dependent variable:
	factor_DL
market_return	0.077
	(0.048)
Constant	0.007***
	(0.002)
Observations	144
R^2	0.017
Adjusted R^2	0.010
Residual Std. Error	0.024 (df = 142)
F Statistic	2.514 (df = 1; 142)
Note:	*p<0.1; **p<0.05; ***p<0.01

Question 4.4

The average monthly return of the momentum factor is higher in both test datasets when stocks are assigned to the long and short portfolios with the Lasso model. The standard deviation is lower in both test datasets with the deep learning approach.

The CAPM regressions on the previous slide show that a momentum strategy based on deep learning predictions is still significantly outperforming the market.

But the momentum factor based on Lasso predictions is clearly the winner, as shown in the figure.