Big Data and AI Strategies

A Practitioner's Introduction to Neural Networks

Global Quantitative and Derivatives Strategy
Peng Cheng, CFA AC
(1-212) 622-5036
peng.cheng@jpmorgan.com
Thomas J Murphy, PhD
(1-212) 270-7377
thomas.x.murphy@jpmchase.com
Marko Kolanovic, PhD
(1-212) 622-3677
marko.kolanovic@jpmorgan.com
J.P. Morgan Securities LLC

- The report aims to demystify neural networks for our readers in a practitioner-friendly way.
- The neural network architecture is explained by comparing it to the familiar linear regression model.
- Using simulated data, we demonstrate how to construct a neural network from scratch in R.
- We then move on to using real world data and examine the correspondence between neural networks and existing, well-known financial models for volatility forecasting:
    - **Feedforward neural network** vs. **ARCH**
    - **Recurrent neural network** vs. **GARCH(1, 1)**
    - **Long short term memory network** vs. **GARCH(p, q)**
- Finally LSTM is used to forecast volatility of S&P 500 and EURUSD, and its performance is compared against GARCH(1, 1).

See page 22 for analyst certification and important disclosures.

Introduction

In this report we aim to demystify neural network for our readers by putting it in a more familiar financial context. When there is already an abundance of literature on neural networks available, why the need for another report?

Compared to other fields in which neural networks have been widely applied, the problems we face in finance are relatively unique, and therefore require separate treatments. We summarize below what are, in our view, the unique challenges investors face when learning about neural networks.

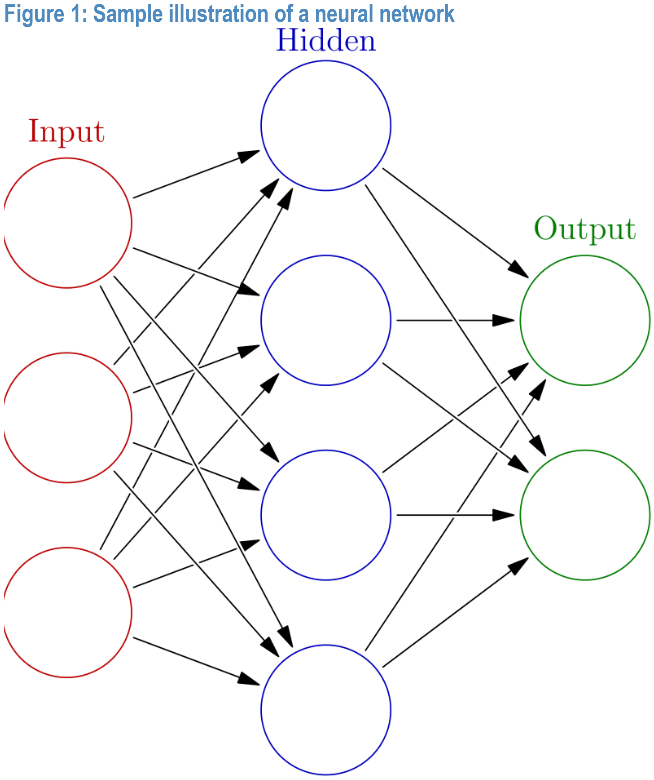
Classification vs. regression

Neural networks are most often discussed in the context of classification, where y variables are either Boolean or categorical. The vast majority of the interesting problems in finance are concerned with continuous variables and are therefore regression problems.

Financial models are not expressed as flow charts

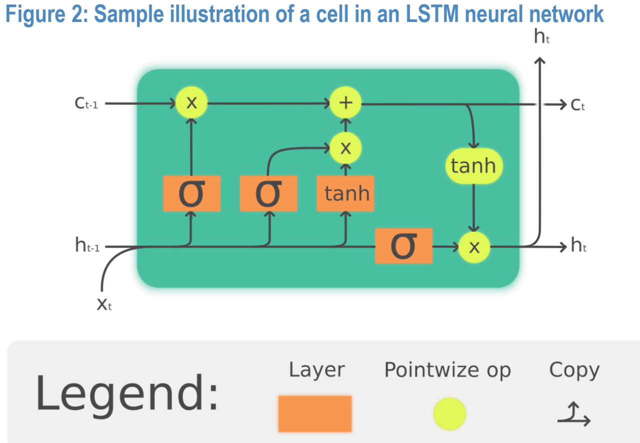
Neural networks are almost always illustrated by graphs seen in Figure 1 and Figure 2, which are unfamiliar to investors since financial models are rarely expressed as such.

Figure 1: Sample illustration of a neural network



Source: Wikipedia/Glosser.ca

Figure 2: Sample illustration of a cell in an LSTM neural network



Source: Wikipedia/Guillaume Chevalier

Unfamiliar terminology

In addition, terms such as **activation function, bias, forward** and **backpropagation** are not commonly found in our glossary, even though very similar concepts already exist in finance.

Lack of working examples

While our colleagues demonstrated the many challenges that neural networks pose when applied to forecasting equity return, finance-related examples, especially the ones that demonstrate predictive power, are few and far between.

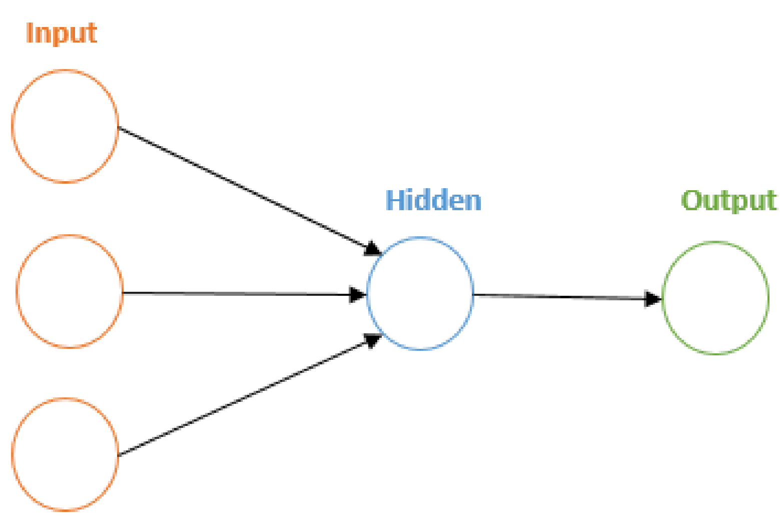
The goal of this report is to introduce neural networks in a practitioner-friendly way, taking into account the points raised above. We first explain the neural network architecture by comparing it to the familiar linear regression model. We then demonstrate how to construct a neural network from scratch in R using simulated data. Finally we move on to real world data and examine the correspondence between neural networks and existing, well-known financial models for volatility forecasting.

Revisiting the Neural Network Architecture

Linear model as a special case of neural network

How would we interpret the graphical representation of a neural network? We start out by showing that a linear regression model can be represented similarly. For instance, the model $y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$ can be seen as a neural network with one hidden layer and one hidden neuron, as shown in Figure 3:

Figure 3: A neural network with one hidden neuron and one hidden layer

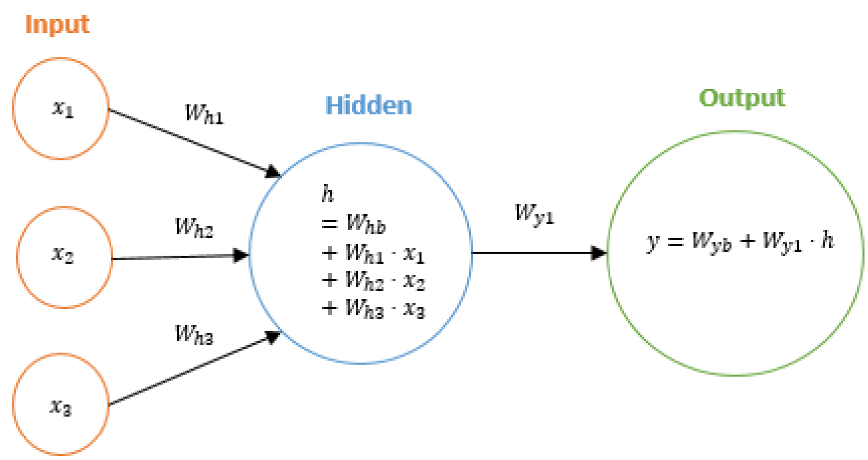


Source: J.P. Morgan

The **input layer** nodes (or **neurons**) are explanatory variables. The three nodes indicate there are three explanatory variables, i.e. x_1, x_2, x_3. We can think of the arrows connecting the nodes as regression coefficients, or **weights**. Each node is a linear combination of the nodes connected into it. We may also add an intercept, a.k.a. **bias**, which may not be explicitly shown, to each node. If we fully annotate the graph, it would look like Figure 4.

Peng Cheng, CFA
(1-212) 622-5036
peng.cheng@jpmorgan.com

Figure 4: Neural network fully annotated



Source: J.P. Morgan

The biases are denoted by W_{hb}, W_{yb}, and the weights are denoted by $W_{h1}, W_{h2}, W_{h3}, W_{y1}$. If we substitute in the hidden node value h into output node value y, we can see that it is exactly the same our linear model $y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$:

$$
\begin{aligned}
y &= W_{yb} + W_{y1} \cdot h \\
&= W_{yb} + W_{y1} \cdot (W_{hb} + W_{h1} \cdot x_1 + W_{h2} \cdot x_2 + W_{h3} \cdot x_3) \\
&= (W_{yb} + W_{y1} \cdot W_{hb}) + \left(W_{y1} \cdot W_{h1}\right)x_1 + \left(W_{y2} \cdot W_{h2}\right)x_2 + \left(W_{y3} \cdot W_{h3}\right)x_3 \\
&= \alpha + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3
\end{aligned}
$$

Where:

$$
\begin{aligned}
\alpha &= W_{yb} + W_{y1} \cdot W_{hb} \\
\beta_1 &= W_{y1} \cdot W_{h1} \\
\beta_2 &= W_{y1} \cdot W_{h2} \\
\beta_3 &= W_{y1} \cdot W_{h3}
\end{aligned}
$$

The procedure for **training** this neural network model is the same as fitting a linear regression, where we choose the biases and weights to minimize a **loss function**, e.g. least squares.

In fact, all neural networks, regardless of the number of neurons or layers, will reduce to a linear regression, if the **activation functions** are linear, which is what we have implicitly chosen for the model above. We now take a closer look at activation functions.

Activation functions

They are functions used to introduce nonlinearity to the neural network. For instance, if we are modeling a probability and would like to restrict our output values y to be between 0 and 1, we can change the output node to:

$$y = \sigma(W_{yb} + W_{y1} \cdot h)$$

Where $\sigma(\cdot)$ is the logistic, a.k.a. **sigmoid** function. It is familiar to us in the context of logistic regression, which is used to model the probability of an event.

$$\sigma(x) = \frac{e^x}{1 + e^x}$$

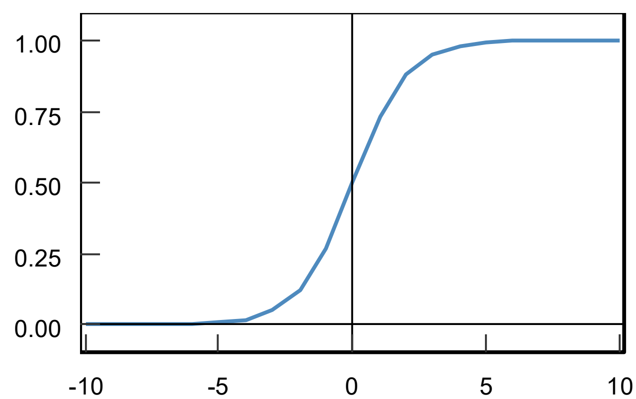
Figure 5 shows how it looks graphically. As we can see, the range of the function is strictly between 0 and 1. Also worth noting is that for x values between -5 and 5, the shape of the function is approximately linear. We will discuss the implications later in the report.

Naturally, activation functions can be applied to any layer besides the output layer. If we apply the sigmoid function to the hidden layer instead, the model will look like:

$$y = W_{yb} + W_{y1} \cdot \sigma(h)$$

In this case, the y variable can still take on all real values.

Figure 5: Sigmoid function



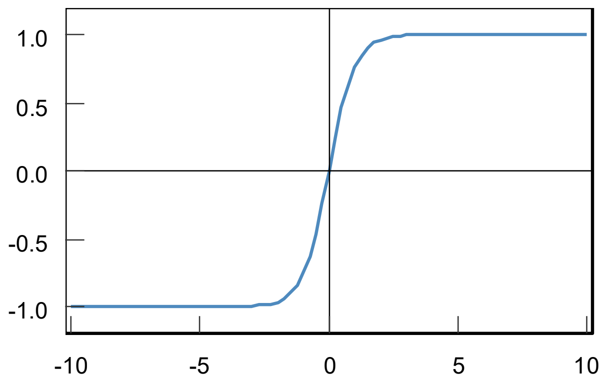
Source: J.P. Morgan

A couple of additional examples that are commonly used can be seen in Figure 6 and Figure 7. They can also be put into economic contexts. **Tanh** (Figure 6) is similar to sigmoid, but the range is between ±1. It may be useful for modelling portfolio weights, restricted between +100% to -100%. **ReLU (**Figure 7) may be used for modelling option strategies given its shape is exactly that of a call option payoff.

Activation functions are at the heart of neural networks. If we only use linear activation functions, then it would be no different from a linear regression. We suggest a couple of best practices for utilizing activation functions. Firstly, for regression problem, we suggest using linear activation functions in order to map the output values to all real values. Secondly, input variables (x_1, x_2, x_3) should be normalized to within reasonable domains of the activation function. For instance, if sigmoid is used, it's best to transform all x variables so that most of them fall between ±5 (e.g. converting them into z-scores). Otherwise, very large or very small values are at risk of being truncated and useful information may be lost.

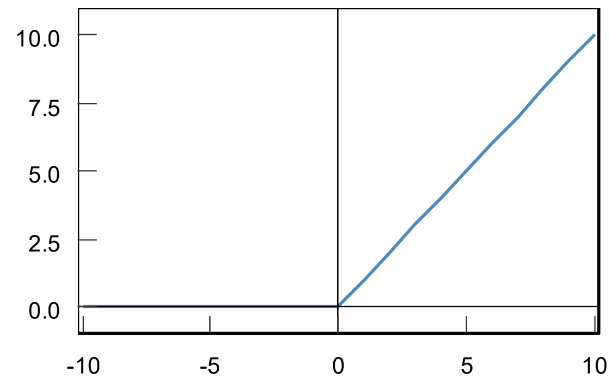
Peng Cheng, CFA
(1-212) 622-5036
peng.cheng@jpmorgan.com

Figure 6: Tanh function



Source: J.P. Morgan

Figure 7: Rectified linear unit (ReLU)



Source: J.P. Morgan

In theory, neural networks are capable of approximating any function regardless of the choice of activation function. However, in practice it is important to be thoughtful about which activation function to use. We will come back to this topic when we demonstrate how to build a neural network from scratch.

Generalization to multiple neurons and multiple layers

Multiple neurons and layers are extensions of the aforementioned framework. In case of multiple neurons, the neural network can be expressed as follows:

$$\begin{aligned} y &= W_{yb} + W_{y1} \cdot \sigma_1(h_1) + W_{y2} \cdot \sigma_2(h_2) + \cdots \\ &= W_{yb} + \sum_i W_{yi} \cdot \sigma_i(h_i) \end{aligned}$$

Where:

$$h_i = W_{ib} + W_{i1} \cdot x_1 + W_{i2} \cdot x_2 + \cdots$$

If one more layer is added on, it can be expressed as the following:

$$\begin{aligned} y = W_{yb} + W_{y1} \cdot \sigma_{11}\left(W_{1b} + \sum_i W_{1i} \cdot \sigma_{1i}(h_{1i})\right) + W_{y2} \\ \cdot \sigma_{12}\left(W_{2b} + \sum_i W_{2i} \cdot \sigma_{2i}(h_{2i})\right) + \cdots \end{aligned}$$

By increasing the neurons/layers, the number of weight parameters increases accordingly, and the neural network becomes a massive nonlinear regression. Our goal remains choosing the weights to minimize the sum of least squares between the model and actual y values.

In Table 1, we show the number of weight parameters assuming we still have three input features, but vary the number of neurons and layers. In building neural networks we would be cautious on deploying too many neurons/layers, especially if the data set is not particularly large. While in-sample fit will inevitably increase, too many parameters increases the risk of overfitting. Additionally, the training

Peng Cheng, CFA
(1-212) 622-5036
peng.cheng@jpmorgan.com

(optimization) process becomes much more computationally intensive and the risk of non-convergence increases.

Table 1: Number of parameters (weights) for a neural network with three input features

Neurons \ Layers	1	2	3	4	5	6	7	8	9	10
1	6	8	10	12	14	16	18	20	22	24
2	11	17	23	29	35	41	47	53	59	65
3	16	28	40	52	64	76	88	100	112	124
4	21	41	61	81	101	121	141	161	181	201
5	26	56	86	116	146	176	206	236	266	296
6	31	73	115	157	199	241	283	325	367	409
7	36	92	148	204	260	316	372	428	484	540
8	41	113	185	257	329	401	473	545	617	689
9	46	136	226	316	406	496	586	676	766	856
10	51	161	271	381	491	601	711	821	931	1041

Source: J.P. Morgan

Building a neural network from scratch

In this section, we show how to build a simple neural network in R from scratch, using only base R functions. [1]

A neural network is made up of two parts. The choice of the number of neurons, hidden layers, and activation functions determines how x variables are transformed into y, given some fixed weights. The transformation process is known as the **forward propagation** step.

Defining the forward propagation is only half of the work, since we want to find out the weights that get us closest to the known y values. The optimization process is known as **backpropagation** for a neural network. Training a neural network involves using an initial guess of the weights, going through the forward propagation and comparing the output with the actual y values, then going back and choosing a new set of weights based on an optimization algorithm (e.g. gradient descent) and repeating the process until the results converge to actual y values within reasonable tolerances.

Backpropagation is a strictly numerical procedure which, although important, carries a lot of technical complexity but little economic intuition. Therefore, we suggest delegating this process to canned optimization routine. In practice, this may require substantial computational power depending on the complexity of the model. For the purpose of our demonstration, the *optim* function built into base R is used.

Below we show how to implement in R the forward propagation step, and use *optim* for the backpropagation.

[1] Parts of the code are modified from https://selbydavid.com/2018/01/09/neural-network by David Selby

Peng Cheng, CFA
(1-212) 622-5036
peng.cheng@jpmorgan.com

We use sigmoid as the activation function for our hidden layer, and linear function for our output layer. The sigmoid is defined in Figure 8.

Figure 8: Define sigmoid function

```
sigmoid <- function(x){
  exp(x)/(1+exp(x))
}
```

Source: J.P. Morgan

We now move on to the forward propagation step. The *cbind(1, ...)* in lines 6 and 8 add the intercepts (biases). The sigmoid function is applied to the hidden node in line 7.

Figure 9: Define forward propagation function

```
 5 fwdprop <- function(x, wh, wy){
 6   h <- cbind(1, x) %*% wh
 7   h <- sigmoid(h) ###hidden layer
 8   output <- cbind(1, h) %*% wy ####output layer
 9   return(list(output = output, h = h))
10 }
```

Source: J.P. Morgan

Similar to OLS, our loss function is defined to be least squares. The *init.w* variable is a vector which contains all the parameters including the intercepts (biases). For now we hard code the first four values to the W_h terms and the last two variables to the W_y terms, for the sake of simplicity.

Figure 10: Define loss function

```
loss.fun <- function(init.w, x, y){
  wh = init.w[1:4]
  wy = init.w[5:6]
  y.hat <- fwdprop(x, wh, wy)$output
  return(sum((y - y.hat)^2))
}
```

Source: J.P. Morgan

The code above constitutes a one layer, one neutron neural network model. It can be expanded relatively easily to accommodate multiple layers and neurons.

Before moving onto the backpropagation step, we first simulate some sample data in order to train the neural network. As opposed to using actual data, we are able to specify the exact data generating process in our simulation. Lines 20 – 25 generate 500 normal random variables with mean zero and standard deviation 0.1. In line 26 we define y as a linear function of x_1, x_2, x_3, and some added noise. The neural network will attempt to use the sigmoid to fit to the linear relationship. This exercise will demonstrate the implication of the choice of activation functions.

Peng Cheng, CFA
(1-212) 622-5036
peng.cheng@jpmorgan.com

Figure 11: Simulate in-sample data

```
set.seed(1) ##seed for random number generator
nobs <- 500
mymean <- 0
mysd <- 0.1
x1 <- rnorm(nobs, mean = mymean, sd = mysd)
x2 <- rnorm(nobs, mean = mymean, sd = mysd)
x3 <- rnorm(nobs, mean = mymean, sd = mysd)
y1 <- 1 + 1*x1 + 0.5*x2 + -0.75*x3 + rnorm(nobs, sd = mysd)
```

Source: J.P. Morgan

Given the data, we use R's built-in optimizer *optim()* to train the neural network. The first argument is the initial guess of the weights. We assign them six normally distributed random variables. The second argument is the loss function, and the subsequent arguments pass the data into the loss function.

Figure 12: Train the model using optim()

```
mysolution <- optim(par = rnorm(6), fn = loss.fun,
  x = cbind(x1, x2, x3), y = y1)
```

Source: J.P. Morgan

We can retrieve the weights from the variable *mysolution* by examining its *par* object. However, we will not be able to recover the true coefficients since we used a sigmoid activation function in our neural network.

To perform an out-of-sample test, we simulate another 500 variables with the same data generating process, but different random variables (lines 31 – 34). In lines 35 – 37, we feed the trained parameters into the forward propagation function and obtain the predictions.

Figure 13: Simulate out-of-sample data

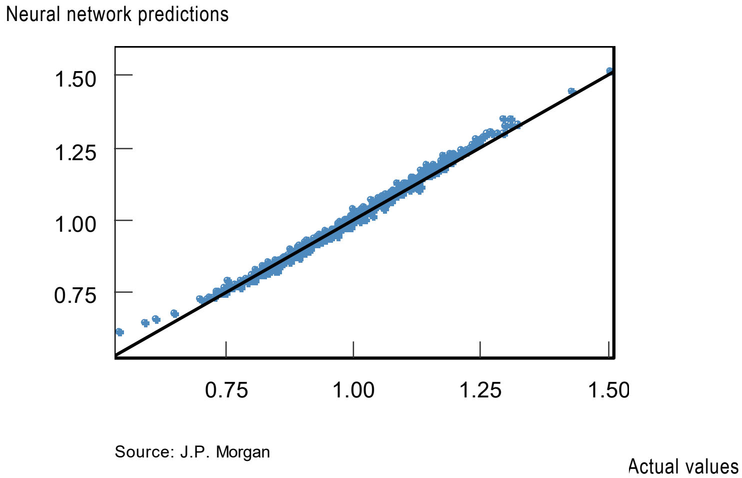
```
31 xx1 <- rnorm(nobs, mean = mymean, sd = mysd)
32 xx2 <- rnorm(nobs, mean = mymean, sd = mysd)
33 xx3 <- rnorm(nobs, mean = mymean, sd = mysd)
34 yy1 <- 1 + 1*xx1 + 0.5*xx2 + -0.75*xx3
35 mypredictions <- fwdprop(x = cbind(xx1, xx2, xx3),
36  wh = mysolution$par[1:4],
37  wy = mysolution$par[5:6])$output
```

Source: J.P. Morgan

The out-of-sample predictions appear reasonable (Figure 14), which is reassuring given the mismatch between the data generating process and our chosen activation function. We observe some nonlinearity for the more extreme values as a result of the sigmoid function.

Peng Cheng, CFA
(1-212) 622-5036
peng.cheng@jpmorgan.com

Figure 14: Predicted vs. actual y values

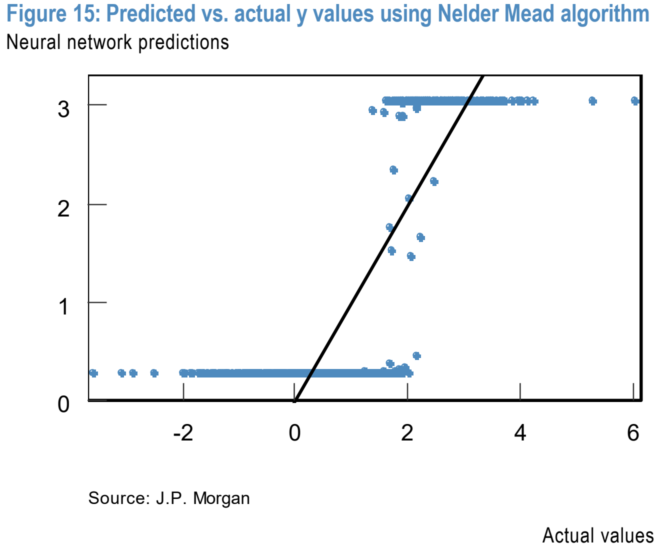


Source: J.P. Morgan

The above exercise is of course an idealized example. In practice, to ensure a reasonable solution, there are many important considerations in the backpropagation process such as the choice of weight initialization, learning rate, etc.

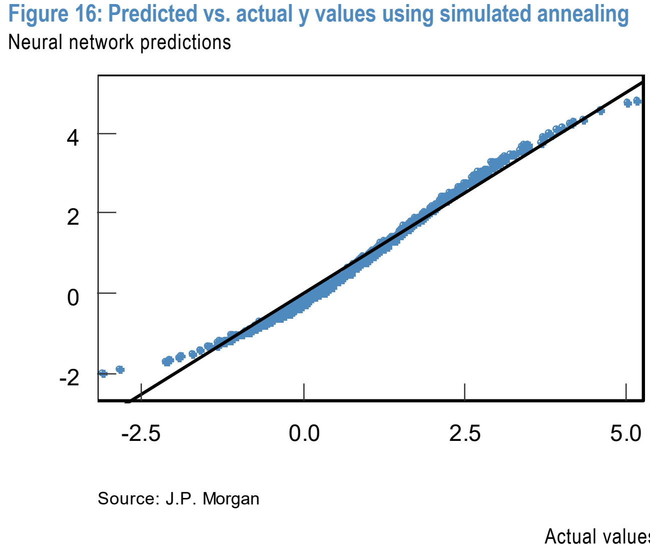
Although the specifics are beyond the scope of this report, we provide a simple illustration by slightly modifying the data. We set the *mysd* value to 1 instead of 0.1 in line 22. By having variables with a wider dispersion and larger magnitude, we end up having a number of observations falling outside of the linear region of the sigmoid function (i.e. between +5 and -5 as shown in Figure 5). As a result, the fit becomes much worse (Figure 15). There are a couple of ways to remedy the problem. One option, as discussed previously, is to normalize our data to make sure that it works well with the activation function. Another option is to use more sophisticated optimization function. Here we choose simulated annealing, which is a stochastic gradient descent method, and can see that the fit is vastly improved (Figure 16).

Figure 15: Predicted vs. actual y values using Nelder Mead algorithm



Source: J.P. Morgan

Figure 16: Predicted vs. actual y values using simulated annealing



Source: J.P. Morgan

Fortunately, in our code, simulated annealing only involves one additional parameter (*method = 'SANN'* in Figure 17).

Figure 17: Extra argument required for simulated annealing

```
mysolution <- optim(par = rnorm(6), fn = loss.fun,
  x = cbind(x1, x2, x3), y = y1, method = 'SANN')
```

Source: J.P. Morgan

Case study: volatility forecasting

Moving on to real world examples, some of the more sophisticated models in finance are also special cases of neural networks. Specifically, we show that there is a one to one correspondence between **ARCH/GARCH** used in volatility forecasting and **feedforward/recurrent neural networks**, respectively. For simplicity, all neural networks shown in this section consist of one hidden layer with one neuron. Adding more neurons and layers does not alter the relationships discussed.

In addition, for demonstration with real data, we use the EURUSD daily data over the last 10 years. The first six rows of the data are seen in Figure 18. *Ret1D* is the log returns of *ClosePrice,* and *Var* is *Ret1D* squared. The *VarLn* variables are *Var* lagged by n days. Here we lag the data for up to five days for our ARCH(5) model below.

Figure 18: Define sigmoid function

Data: head(eurusd)

File

	CloseDate	ClosePrice	Ret1D	Var	VarL1	VarL2	VarL3	VarL4	VarL5
1	2009-08-10	1.4140	-0.0061339072	3.762482e-05	8.529179e-05	6.270020e-06	8.148381e-07	1.926603e-08	1.458829e-04
2	2009-08-11	1.4131	-0.0006366949	4.053804e-07	3.762482e-05	8.529179e-05	6.270020e-06	8.148381e-07	1.926603e-08
3	2009-08-12	1.4211	0.0056453470	3.186994e-05	4.053804e-07	3.762482e-05	8.529179e-05	6.270020e-06	8.148381e-07
4	2009-08-13	1.4277	0.0046335382	2.146968e-05	3.186994e-05	4.053804e-07	3.762482e-05	8.529179e-05	6.270020e-06
5	2009-08-14	1.4215	-0.0043521057	1.894082e-05	2.146968e-05	3.186994e-05	4.053804e-07	3.762482e-05	8.529179e-05
6	2009-08-17	1.4077	-0.0097554853	9.516949e-05	1.894082e-05	2.146968e-05	3.186994e-05	4.053804e-07	3.762482e-05

Source: J.P. Morgan

Feedforward Neural Network ⇔ ARCH

A simple ARCH(p) model is specified as follows, where $\mathcal{N}(\mu, \sigma)$ is the normal distribution. Although one can use ARCH to model asset returns (r_t), here we are mainly interested in volatility forecasting, i.e. σ_t^2.

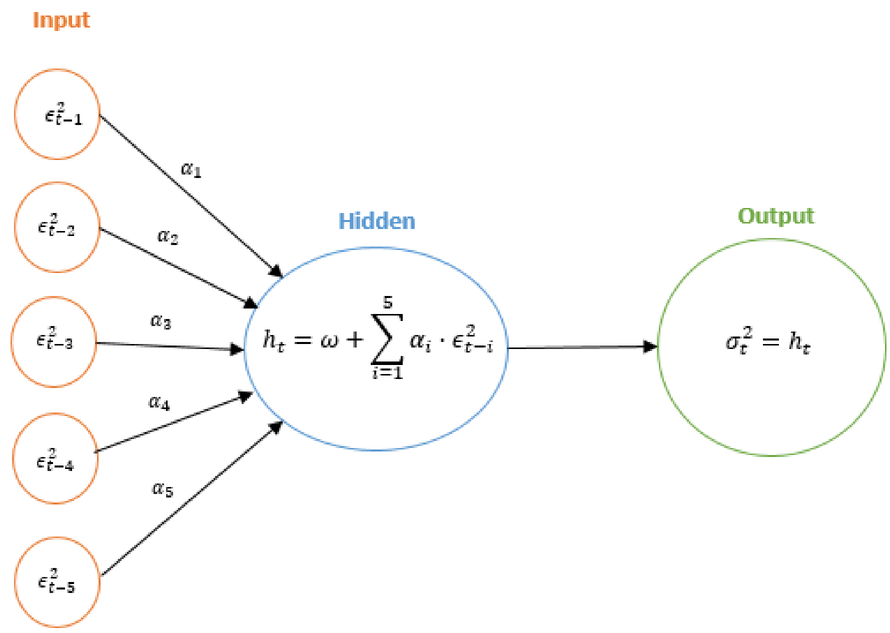
$$r_t = \epsilon_t, \epsilon_t \sim \mathcal{N}(0, \sigma_t)$$

$$\sigma_t^2 = \omega + \alpha_1 \epsilon_{t-1}^2 + \alpha_2 \epsilon_{t-2}^2 + \cdots + \alpha_p \epsilon_{t-p}^2$$

As we can see, the ARCH model is essentially a linear model and can therefore be approximated by a simple neural network (Figure 19). It is also known as a feedforward neural network, in contrast to a recurrent neural network, which we will get to in the next section.

Peng Cheng, CFA
(1-212) 622-5036
peng.cheng@jpmorgan.com

Figure 19: ARCH expressed as a feedforward neural network



Source: J.P. Morgan

To make the neural network consistent with ARCH, we choose the linear activation function for the hidden layer and the identify function for the output layer. Moreover, since ARCH is commonly estimated using maximum likelihood estimation (MLE), we do the same for our loss function. Specifically, we set the loss function to minimize negative sum of log likelihood, as seen below, where $f(\mu, \sigma)$ is the likelihood function of the normal distribution:

$$\sum_i -\log(f(y_t, \sigma_t))$$

Compare the code for the classical ARCH and neural network expression:

Figure 20: Classical ARCH definition in R

```
myARCHfit <- function(params){
 sigma2t <- with(eurusd, params[1] +
  params[2] * VarL1 +
  params[3] * VarL2 +
  params[4] * VarL3 +
  params[5] * VarL4 +
  params[6] * VarL5)
 sigmat <- sqrt(sigma2t)
 log.likelihood <- log(dnorm(eurusd$Ret1D, 0, sigmat))
 return(-1 * sum(log.likelihood))
}

init.params <- c(0, rep(0.2, 5))
arch <- optim(par = init.params, fn = myARCHfit)
```

Source: J.P. Morgan

Peng Cheng, CFA
(1-212) 622-5036
peng.cheng@jpmorgan.com

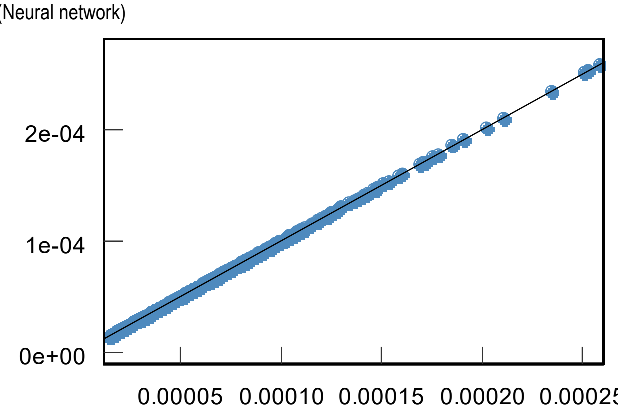
Figure 21: ARCH expressed as a feedforward neural network in R

```
actfun <- function(x) {x} ###linear activation function
fwdprop <- function(x, w){
 h <- actfun(cbind(1, x) %*% (w))
 y <- h ###output layer: identity function
 list(output = y)
}

loss.fun <- function(init.w, x, y){
 y.hat <- fwdprop(x, init.w)$output
 return(-sum(log(dnorm(y, 0, sqrt(y.hat)))))
}
init.guess <- c(0, rep(0.2, 5))

arch.nn <- optim(par = init.guess, fn = loss.fun,
 x = as.matrix(eurusd[, c('VarL1', 'VarL2', 'VarL3',
 'VarL4', 'VarL5')]), y = eurusd[, 'Ret1D'])
```

Source: J.P. Morgan

In Figure 22 and Table 2 we compare the output and parameter values, and find the two models to be identical.

Figure 22: ARCH vs. neural network output values



Source: J.P. Morgan

Table 2: ARCH vs. neural network parameter values

	ARCH	NN
omega	0.000014	0.000014
alpha1	0.065149	0.065149
alpha2	0.277980	0.277980
alpha3	0.186251	0.186251
alpha4	0.195344	0.195344
alpha5	0.087819	0.087819

Source: J.P. Morgan

Recurrent Neural Network⇔ GARCH(1, 1)

It is generally accepted that ARCH with longer lags tends to produce better results. GARCH(1, 1) is a parsimonious way of parametrizing ARCH(∞).

$$r_t = \epsilon_t, \epsilon_t \sim \mathcal{N}(0, \sigma_t)$$

$$\sigma_t^2 = \omega + \alpha\epsilon_{t-1}^2 + \beta\sigma_{t-1}^2$$

By expanding the σ_{t-1}^2 term we can see it contains all the previous ϵ_{t-i} terms, and the weights of ϵ_{t-i} decay exponentially with i.

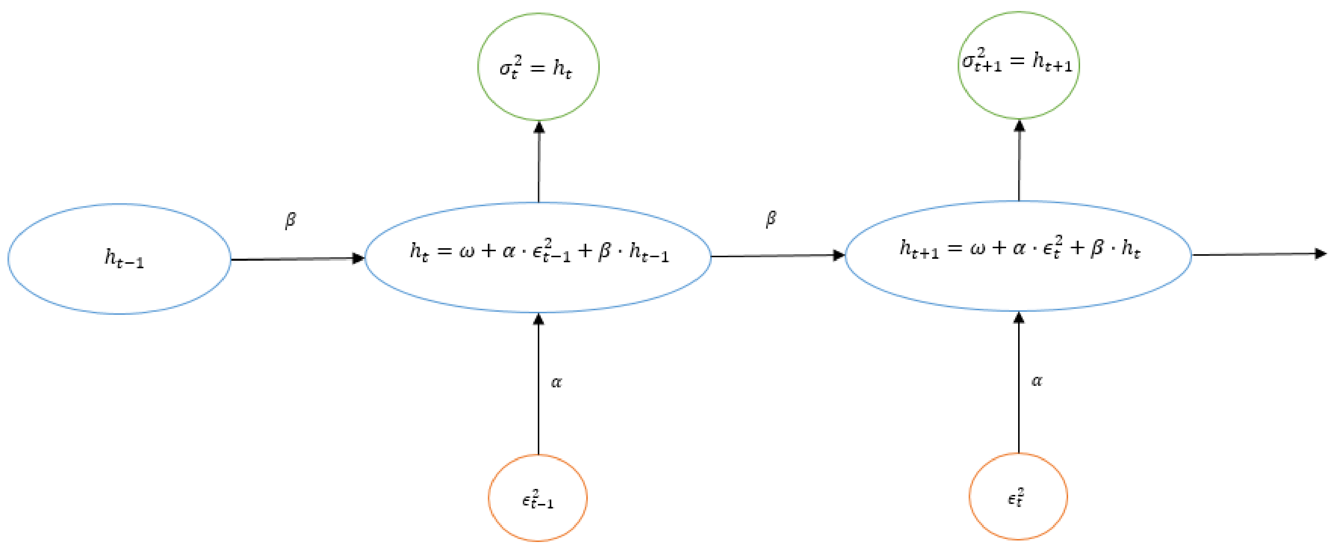
$$\begin{aligned}\sigma_t^2 &= \omega + \alpha\epsilon_{t-1}^2 + \beta\sigma_{t-1}^2 \\ &= \omega + \alpha\epsilon_{t-1}^2 + \beta(\omega + \alpha\epsilon_{t-2}^2 + \beta(\omega + \alpha\epsilon_{t-3}^2 + \beta(\omega + \alpha\epsilon_{t-4}^2 + \cdots)))\end{aligned}$$

Peng Cheng, CFA
(1-212) 622-5036
peng.cheng@jpmorgan.com

$$= \sum_{i=0}^{\infty} \beta^i (\omega + \alpha \epsilon_{t-1-i}^2)$$

While a feedforward neural network is not able to capture the recursive structure, a recurrent neural network (RNN) does exactly that. In Figure 23, we show what happens in an RNN. The horizontal arrows no longer indicate connections between neurons, but between one time period and the next within a single neuron. The input and output layers remain the same as a feedforward neural network, but the h_t in the hidden layer will take h_{t-1} as an additional input. In other words, h_t plays the role of σ_t^2 in the GARCH model.

Figure 23: GARCH expressed as a recurrent neural network



Source: J.P. Morgan

Again compare the classical GARCH vs. the RNN expression:

Figure 24: Classical GARCH definition in R

```
myGARCHfit <- function(params){
  omega <- params[1]
  alphaL1 <- params[2]
  betaL1 <- params[3]
  sigma2t <- rep(mean(eurusd$VarL1), nrow(eurusd))
  for (i in 2:nrow(eurusd)){
    sigma2t[i] <- omega + alphaL1 * eurusd$VarL1[i] +
     betaL1 * sigma2t[i-1]
  }
  sigmat <- sqrt(sigma2t)
  log.likelihood <- log(dnorm(eurusd$Ret1D, 0, sigmat))
  return(-1 * sum(log.likelihood))
}

init.params <- c(0, rep(0.5, 2))
garch <- optim(par = init.params, fn = myGARCHfit)
```

Source: J.P. Morgan

Figure 25: GARCH expressed as a recurrent neural network in R

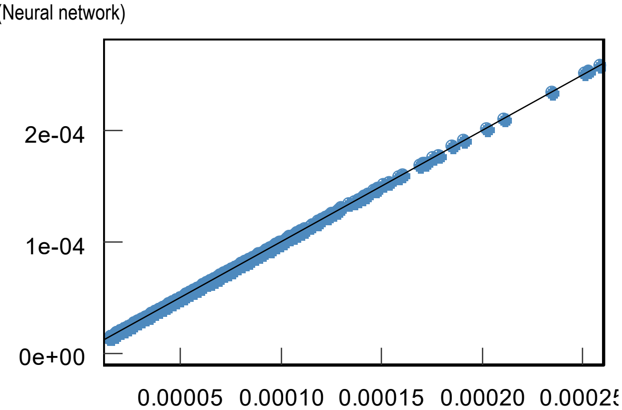
```
actfun <- function(x) {x} ##linear activation function
rnn.fwdprop <- function(x, w){
 h <- matrix(nrow = nrow(x))
 h[1, 1] <- mean(x)
 for (k in 2:nrow(h)){
  h[k, 1] <- actfun(cbind(1, x[k, 1], h[k-1, 1]) %*% w)
 }
 y <- h ###identity function output
 list(output = y)
}

loss.fun <- function(init.w, x, y){
 y.hat <- rnn.fwdprop(x, init.w)$output
 return(-sum(log(dnorm(y, 0, sqrt(y.hat)))))
}

init.guess <- c(0, rep(0.5, 2))
garch.nn <- optim(par = init.guess, fn = loss.fun,
 x = as.matrix(eurusd[, c('VarL1')]),
 y = eurusd[, 'Ret1D'])
```

Source: J.P. Morgan

The results are shown below and they produce the same output as expected.

Figure 26: GARCH vs. RNN output values

σ_t^2 (Neural network)



Source: J.P. Morgan

σ_t^2 (GARCH)

Table 3: GARCH vs. RNN parameter values

	GARCH	RNN
omega	4.84E-08	4.84E-08
alpha	0.026377	0.026377
beta	0.972016	0.972016

Source: J.P. Morgan

LSTM ⇔ GARCH(p, q)

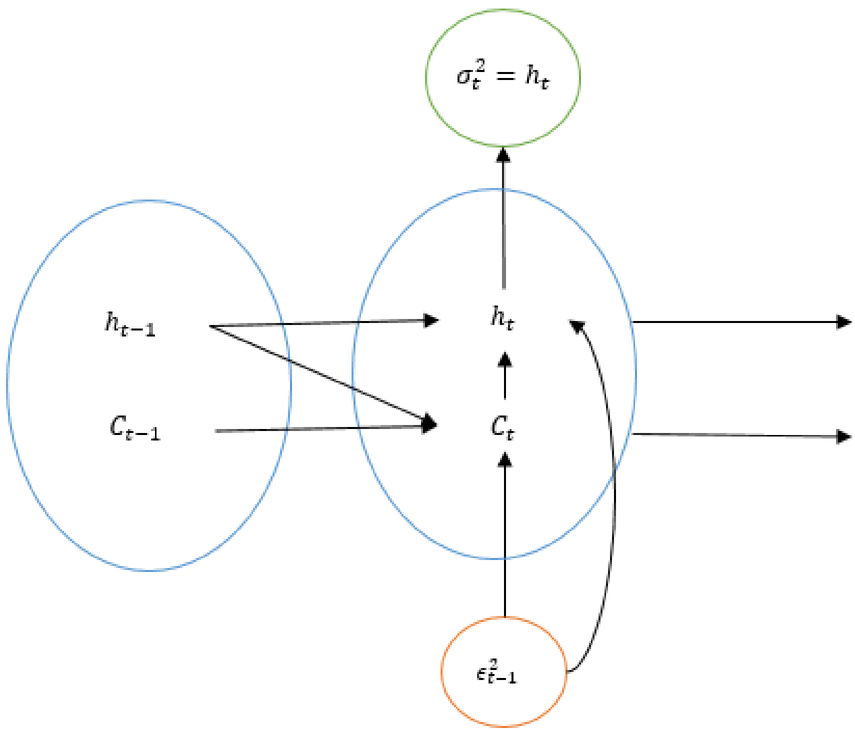
While GARCH(1, 1) is a parsimonious way of parameterizing long lags, long term memory still tends to get lost. Recall in GARCH(1, 1), the weights for ϵ_t^2 decay at a rate of β every period. The importance of the more distant historical observations will decay to close to 0 relatively quickly. For instance, in our example above, with $\beta = 0.972016$, the weights for observations more than nine months old will be close to 0.

$$\begin{aligned}\sigma_t^2 &= \omega + \alpha\epsilon_{t-1}^2 + \beta\sigma_{t-1}^2 \\ &= \omega + \alpha\epsilon_{t-1}^2 + \beta(\omega + \alpha\epsilon_{t-2}^2 + \beta(\omega + \alpha\epsilon_{t-3}^2 + \beta(\omega + \alpha\epsilon_{t-4}^2 + \cdots)))\end{aligned}$$

$$= \sum_{i=0}^{\infty} \beta^i (\omega + \alpha \epsilon_{t-1-i}^2)$$

One way to remedy the problem is to introduce more lagged terms, such as GARCH(2, 2). When researchers encountered the same problem with RNN, the solution proposed was Long Short Term Memory (LSTM), which is roughly illustrated in Figure 27. In addition to the h_t time series, LSTM maintains an alternative time series C_t which allows for historical information to be retained at an alternative, perhaps slower, decay rate. In addition to the usual RNN inputs, h_t is also a function of C_t.

Figure 27: ARCH expressed as a feedforward neural network



Source: J.P. Morgan

Below we implement a basic LSTM network in the volatility forecasting framework[2]:

$$r_t = \epsilon_t, \epsilon_t \sim \mathcal{N}(0, \sigma_t)$$

$$\begin{aligned}
\sigma_t^2 &= o_t C_t \\
C_t &= f_t C_{t-1} + i_t(\omega + \alpha \epsilon_{t-1}^2 + \beta \sigma_{t-1}^2) \\
f_t &= \sigma\left(w_f + a_f \epsilon_{t-1}^2 + b_f \sigma_{t-1}^2\right) \\
i_t &= \sigma(w_i + a_i \epsilon_{t-1}^2 + b_i \sigma_{t-1}^2) \\
o_t &= \sigma(w_o + a_o \epsilon_{t-1}^2 + b_o \sigma_{t-1}^2)
\end{aligned}$$

[2] A more thorough explanation of LSTM can be found here: https://colah.github.io/posts/2015-08-Understanding-LSTMs/ by Christopher Olah

Peng Cheng, CFA
(1-212) 622-5036
peng.cheng@jpmorgan.com

While the model may appear intimidating at first, the intuition is relatively straight-forward. First note that f_t and i_t are both sigmoid functions and therefore their values are restricted to between 0 and 1. C_t therefore is a weighted sum between its previous period value (C_{t-1}), i.e. long term memory, and a GARCH model ($\omega + \alpha\epsilon_{t-1}^2 + \beta\sigma_{t-1}^2$), i.e. short term memory. If $i_t = 1, f_t = 0, o_t = 1$, then the LSTM simplifies into an RNN. By making f_t, i_t, o_t (known as **gates**) dependent on ϵ^2 and σ^2 at each point in time, the weights attached to historical observations are much more varied, rather than simply decaying exponentially. If we expand the C_t term, we can see that it contains infinite lags of ϵ_t^2 and σ_t^2. Therefore, LSTM appears to be a more generalized version of GARCH with infinite lags.

As before, we present two ways of implementing the forward propagation step and loss function for LSTM-GARCH model in Figure 28 and Figure 29.

Figure 28: LSTM-GARCH implementation in R type I

```
actfun <- function(x) {x} ##linear activation function
sigmoid <- function(x){exp(x)/(1+exp(x))}
lstm.fwdprop <- function(x, w){
 omega <- w[1]; alpha <- w[2]; beta <- w[3]
 wf <- w[4]; af <- w[5]; bf <- w[6]
 wi <- w[7]; ai <- w[8]; bi <- w[9]
 wo <- w[10]; ao <- w[11]; bo <- w[12]

 h <- matrix(nrow = nrow(x))
 h[1, 1] <- mean(x)

 ot <- rep(0, nrow(x))
 ft <- rep(0, nrow(x))
 it <- rep(0, nrow(x))
 Ct <- matrix(nrow = nrow(x))

 for (k in 2:nrow(h)){
  ot[k] <- sigmoid(wo + ao * xdf$VarL1[k] +
   bo * sigma2t[k-1])
  ft[k] <- sigmoid(wf + af * xdf$VarL1[k] +
   bf * sigma2t[k-1])
  it[k] <- sigmoid(wi + ai * xdf$VarL1[k] +
   bi * sigma2t[k-1])
  Ct[k] <- ft[k] * Ct[k-1] +
   it[k] * actfun(omega + alpha * xdf$VarL1[k] +
   beta * sigma2t[k-1])
  h[k, 1] <- ot[k] * actfun(Ct[k])
 }
 y <- h ###identity function output
 list(output = y)
}

loss.fun <- function(init.w, x, y){
 y.hat <- lstm.fwdprop(x, init.w)$output
 return(-sum(log(dnorm(y, 0, sqrt(y.hat)))))
}
```

Source: J.P. Morgan

Figure 29: LSTM-GARCH implementation in R type II

```
sigmoid <- function(x){exp(x)/(1+exp(x))}
myLSTM <- function(params, xdf){
 omega <- params[1]; alpha <- params[2]; beta <- params[3]
 wf <- params[4]; af <- params[5]; bf <- params[6]
 wi <- params[7]; ai <- params[8]; bi <- params[9]
 wo <- params[10]; ao <- params[11]; bo <- params[12]

 sigma2t <- rep(mean(xdf$VarL1), nrow(xdf))
 ot <- rep(0, nrow(xdf))
 ft <- rep(0, nrow(xdf))
 it <- rep(0, nrow(xdf))
 Ct <- rep(0, nrow(xdf))

 for (i in 2:nrow(xdf)){
  ot[i] <- sigmoid(wo + ao * xdf$VarL1[i] +
   bo * sigma2t[i-1])
  ft[i] <- sigmoid(wf + af * xdf$VarL1[i] +
   bf * sigma2t[i-1])
  it[i] <- sigmoid(wi + ai * xdf$VarL1[i] +
   bi * sigma2t[i-1])
  Ct[i] <- ft[i] * Ct[i-1] +
   it[i] * (omega + alpha * xdf$VarL1[i] +
   beta * sigma2t[i-1])
  sigma2t[i] <- ot[i] * Ct[i]
 }
 sigmat <- sqrt(sigma2t)
 log.likelihood <- log(dnorm(xdf$Ret1D, 0, sigmat))
 return(-1 * sum(log.likelihood))
}
```

Source: J.P. Morgan

Training an LSTM network becomes considerably more computationally intensive, and therefore we use a more robust optimization library *Rsolnp* which supports parallel processing. Without going into too much detail, the code in Figure 29 trains the LSTM network on historical daily EURUSD data.

Figure 30: Optimization function for training the LSTM

```
clusterExport(workers, c('sigmoid', 'myLSTM'))
system.time(lstm <- gosolnp(fun = myLSTM, xdf = mydf,
 LB = c(0, 0.0, 0.0, rep(c(-5, -1.5e5, -1.5e5), 3)),
 UB = c(0.05, 1, 1, rep(c(5, 1.5e5, 1.5e5), 3)),
 cluster = workers))
```

Source: J.P. Morgan

LSTM vs. GARCH(1, 1)

Although it has been shown that GARCH specification with longer lags does not significantly outperform GARCH(1, 1)[3], the weights in LSTM are much more flexible. To our knowledge, there is no GARCH type model with such a rich parametrization of weights[4]. Therefore, we are curious to see how LSTM performs relative to GARCH.

---

[3] A forecast comparison of volatility models: Does anything beat a GARCH(1, 1)? – P. H. Hansen, A. Lunde, Journal of Applied Econometrics, Mar 2005

[4] A list of GARCH family of models: https://vlab.stern.nyu.edu/docs/volatility by Robert Engle

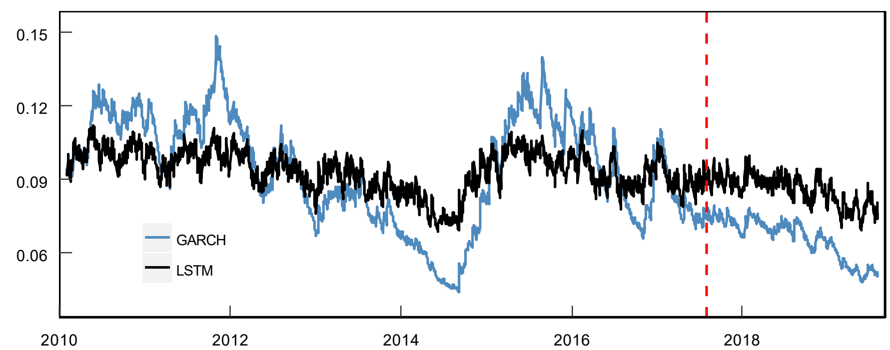
Peng Cheng, CFA
(1-212) 622-5036
peng.cheng@jpmorgan.com

Using the last ten years' daily data on EURUSD exchange rate and S&P 500, we benchmark the performance of LSTM against GARCH(1, 1) for forecasting day-ahead volatility.

Specifically, we divide the historical data into **in-sample (first eight years)** and **out-of-sample (most recent two years)**. Both models are fitted to the in-sample data, then the parameters are used to make predictions on out-of-sample data. When analyzing the results below, we leave out the first six months of in-sample data as the 'burn-in' period.

In Figure 31 and Figure 32 we show the LSTM and GARCH(1, 1) outputs for each asset, both in- and out-of-sample (delineated by the red dotted line). Interestingly, we see that for EURUSD the LSTM outputs have highly persistent long term memory. This is also confirmed by inspecting the gate values (Figure 33): f_t which is the importance attached to long term memory is very close to 1, and i_t, the importance of short term memory, is close to 0.

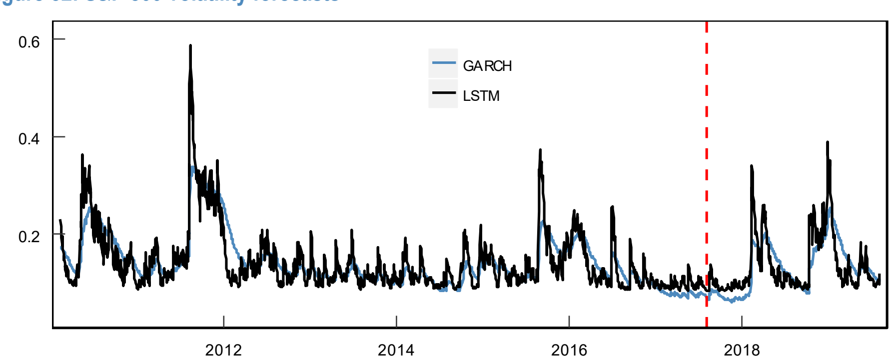
Figure 31: EURUSD volatility forecasts



Source: J.P. Morgan

For S&P 500, the picture is quite different. LSTM and GARCH output very similar values. Again by inspecting the gate values, our observations are confirmed (Figure 34): long term memory weights are close to 0 and short term memory weights are close to 1.

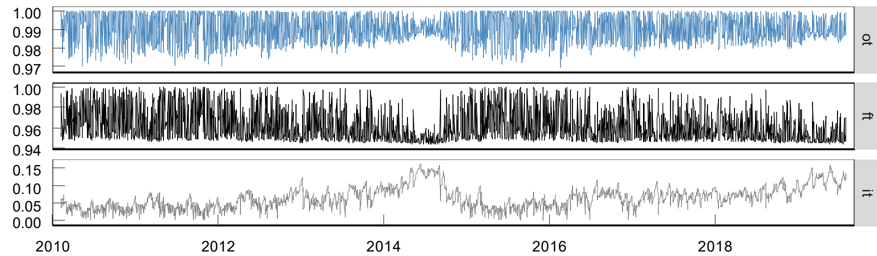
Figure 32: S&P 500 volatility forecasts



Source: J.P. Morgan

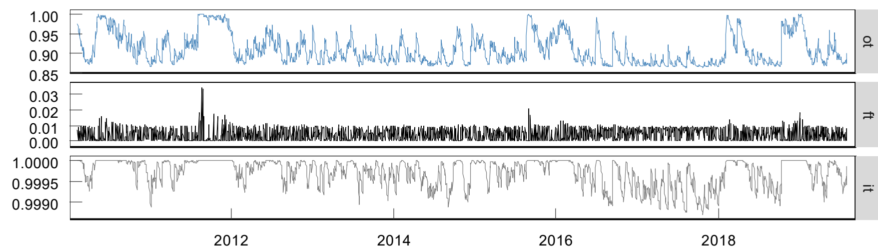
Peng Cheng, CFA
(1-212) 622-5036
peng.cheng@jpmorgan.com

Figure 33: LSTM gate values for EURUSD



Source: J.P. Morgan

Figure 34: LSTM gate values for S&P 500



Source: J.P. Morgan

We consider several metrics to quantify the performance: sum of absolute deviation (SAD), sum of squared errors (SSE), R^2 of regressions between actual and output values, and finally, Akaike information criterion (AIC). SAD takes the absolute value of errors and therefore is robust to outliers. On the other hand, SSE and R^2 take the square of errors and are overweight large errors. AIC measures the sum of likelihood (which is our loss function), after adjusting for the number of parameters. Since GARCH has only 3 parameters and LSTM has 12, such an adjustment is meaningful, in our view.

The results are shown in Table 4. The preferred model is one with smaller out-of-sample values in all three metrics. In general, we find LSTM performs in-line to slightly better than GARCH(1, 1) in forecasting out-of-sample S&P 500 volatility. On the other hand, GARCH(1, 1) does better in forecasting out-of-sample EURUSD volatility.

Peng Cheng, CFA
(1-212) 622-5036
peng.cheng@jpmorgan.com

Table 4: Performance metrics comparison

	S&P 500		EURUSD	
SAD	**In-sample**	**Out-of-sample**	**In-sample**	**Out-of-sample**
GARCH(1, 1)	0.183	0.0441	0.0728	0.00778
LSTM	0.182	0.0447	0.0726	0.01120
SSE	**In-sample**	**Out-of-sample**	**In-sample**	**Out-of-sample**
GARCH(1, 1)	8.90E-05	1.94E-05	7.78E-06	3.13E-02
LSTM	8.31E-05	1.86E-05	7.90E-06	6.48E-02
R2	**In-sample**	**Out-of-sample**	**In-sample**	**Out-of-sample**
GARCH(1, 1)	0.105	0.051	0.0420	0.0210
LSTM	0.170	0.100	0.0347	0.0090
AIC	**In-sample**	**Out-of-sample**	**In-sample**	**Out-of-sample**
GARCH(1, 1)	-13518	-3490	-15558	-4296
LSTM	-13523	-3489	-15450	-4193

Source: J.P. Morgan

Conclusion

In this report we aim to bridge the gap between neural networks and commonly used models in finance, progressively from linear regression to GARCH(p, q). These parallels not only help us understand the inner workings of neural networks, but also allow us to better judge the potential effectiveness of neural networks. For instance, RNN and LSTM appear useful for volatility forecasting, since they share key characteristics with GARCH, which have been proven to work well in volatility forecasting.

At the same time, this exercise also offers potential ways to improve the existing finance model. The rich parametrization of LSTM weights, for example, may be incorporated into GARCH to capture the long memory property of certain volatility time series. As we have shown above, the LSTM enhancements show promising signs in out-of-sample testing and may be of further research interest.

Analyst Certification: All authors named within this report are research analysts unless otherwise specified. The research analyst(s) denoted by an "AC" on the cover of this report certifies (or, where multiple research analysts are primarily responsible for this report, the research analyst denoted by an "AC" on the cover or within the document individually certifies, with respect to each security or issuer that the research analyst covers in this research) that: (1) all of the views expressed in this report accurately reflect his or her personal views about any and all of the subject securities or issuers; and (2) no part of any of the research analyst's compensation was, is, or will be directly or indirectly related to the specific recommendations or views expressed by the research analyst(s) in this report. For all Korea-based research analysts listed on the front cover, if applicable, they also certify, as per KOFIA requirements, that their analysis was made in good faith and that the views reflect their own opinion, without undue influence or intervention.

Important Disclosures

This report is a product of the research department's Global Equity Derivatives and Delta One Strategy group. Views expressed may differ from the views of the research analysts covering stocks or sectors mentioned in this report. Structured securities, options, futures and other derivatives are complex instruments, may involve a high degree of risk, and may be appropriate investments only for sophisticated investors who are capable of understanding and assuming the risks involved. Because of the importance of tax considerations to many option transactions, the investor considering options should consult with his/her tax advisor as to how taxes affect the outcome of contemplated option transactions.

Company-Specific Disclosures: Important disclosures, including price charts and credit opinion history tables, are available for compendium reports and all J.P. Morgan–covered companies by visiting https://www.jpmm.com/research/disclosures, calling 1-800-477-0406, or e-mailing research.disclosure.inquiries@jpmorgan.com with your request. J.P. Morgan's Strategy, Technical, and Quantitative Research teams may screen companies not covered by J.P. Morgan. For important disclosures for these companies, please call 1-800-477-0406 or e-mail research.disclosure.inquiries@jpmorgan.com.

Explanation of Equity Research Ratings, Designations and Analyst(s) Coverage Universe:
J.P. Morgan uses the following rating system: Overweight [Over the next six to twelve months, we expect this stock will outperform the average total return of the stocks in the analyst's (or the analyst's team's) coverage universe.] Neutral [Over the next six to twelve months, we expect this stock will perform in line with the average total return of the stocks in the analyst's (or the analyst's team's) coverage universe.] Underweight [Over the next six to twelve months, we expect this stock will underperform the average total return of the stocks in the analyst's (or the analyst's team's) coverage universe.] Not Rated (NR): J.P. Morgan has removed the rating and, if applicable, the price target, for this stock because of either a lack of a sufficient fundamental basis or for legal, regulatory or policy reasons. The previous rating and, if applicable, the price target, no longer should be relied upon. An NR designation is not a recommendation or a rating. In our Asia (ex-Australia and ex-India) and U.K. small- and mid-cap equity research, each stock's expected total return is compared to the expected total return of a benchmark country market index, not to those analysts' coverage universe. If it does not appear in the Important Disclosures section of this report, the certifying analyst's coverage universe can be found on J.P. Morgan's research website, www.jpmorganmarkets.com.

J.P. Morgan Equity Research Ratings Distribution, as of July 06, 2019

	Overweight (buy)	Neutral (hold)	Underweight (sell)
J.P. Morgan Global Equity Research Coverage	45%	41%	14%
IB clients*	52%	49%	36%
JPMS Equity Research Coverage	42%	44%	14%
IB clients*	76%	65%	56%

*Percentage of subject companies within each of the "buy," "hold" and "sell" categories for which J.P. Morgan has provided investment banking services within the previous 12 months.
For purposes only of FINRA ratings distribution rules, our Overweight rating falls into a buy rating category; our Neutral rating falls into a hold rating category; and our Underweight rating falls into a sell rating category. Please note that stocks with an NR designation are not included in the table above. This information is current as of the end of the most recent calendar quarter.

Equity Valuation and Risks: For valuation methodology and risks associated with covered companies or price targets for covered companies, please see the most recent company-specific research report at http://www.jpmorganmarkets.com, contact the primary analyst or your J.P. Morgan representative, or email research.disclosure.inquiries@jpmorgan.com. For material information about the proprietary models used, please see the Summary of Financials in company-specific research reports and the Company Tearsheets, which are available to download on the company pages of our client website, http://www.jpmorganmarkets.com. This report also sets out within it the material underlying assumptions used.

Analysts' Compensation: The research analysts responsible for the preparation of this report receive compensation based upon various factors, including the quality and accuracy of research, client feedback, competitive factors, and overall firm revenues.

Other Disclosures

J.P. Morgan is a marketing name for investment banking businesses of JPMorgan Chase & Co. and its subsidiaries and affiliates worldwide.

All research reports made available to clients are simultaneously available on our client website, J.P. Morgan Markets. Not all research content is redistributed, e-mailed or made available to third-party aggregators. For all research reports available on a particular stock, please contact your sales representative.

Any data discrepancies in this report could be the result of different calculations and/or adjustments.

Options and Futures related research: If the information contained herein regards options or futures related research, such information is available only to persons who have received the proper options or futures risk disclosure documents. Please contact your J.P. Morgan Representative or visit https://www.theocc.com/components/docs/riskstoc.pdf for a copy of the Option Clearing Corporation's Characteristics and Risks of Standardized Options or http://www.finra.org/sites/default/files/Security_Futures_Risk_Disclosure_Statement_2018.pdf for a copy of the Security Futures Risk Disclosure Statement.

Principal Trading: J.P. Morgan trades or may trade as principal in the derivatives or the debt securities (or related derivatives) that are the subject of this report.

Private Bank Clients: Where you are receiving research as a client of the private banking businesses offered by JPMorgan Chase & Co. and its subsidiaries ("J.P. Morgan Private Bank"), research is provided to you by J.P. Morgan Private Bank and not by any other division of J.P. Morgan, including but not limited to the J.P. Morgan corporate and investment bank and its research division.

Legal entity responsible for the production of research: The legal entity identified below the name of the Reg AC research analyst who authored this report is the legal entity responsible for the production of this research. Where multiple Reg AC research analysts authored this report with different legal entities identified below their names, these legal entities are jointly responsible for the production of this research.

Legal Entities Disclosures
U.S.: JPMS is a member of NYSE, FINRA, SIPC and the NFA. JPMorgan Chase Bank, N.A. is a member of FDIC. **Canada**: J.P. Morgan Securities Canada Inc. is a registered investment dealer, regulated by the Investment Industry Regulatory Organization of Canada and the Ontario Securities Commission and is the participating member on Canadian exchanges. **U.K.**: JPMorgan Chase N.A., London Branch, is authorised by the Prudential Regulation Authority and is subject to regulation by the Financial Conduct Authority and to limited regulation by the Prudential Regulation Authority. Details about the extent of our regulation by the Prudential Regulation Authority are available from J.P. Morgan on request. J.P. Morgan Securities plc (JPMS plc) is a member of the London Stock Exchange and is authorised by the Prudential Regulation Authority and regulated by the Financial Conduct Authority and the Prudential Regulation Authority. Registered in England & Wales No. 2711006. Registered Office 25 Bank Street, London, E14 5JP. **Germany**: This material is distributed in Germany by J.P. Morgan Securities plc, Frankfurt Branch which is regulated by the Bundesanstalt für Finanzdienstleistungsaufsich and also by J.P. Morgan AG (JPM AG) which is a member of the Frankfurt stock exchange and is regulated by the Federal Financial Supervisory Authority (BaFin), JPM AG is a company incorporated in the Federal Republic of Germany with registered office at Taunustor 1, 60310 Frankfurt am Main, the Federal Republic of Germany. **South Africa**: J.P. Morgan Equities South Africa Proprietary Limited is a member of the Johannesburg Securities Exchange and is regulated by the Financial Services Board. **Hong Kong**: J.P. Morgan Securities (Asia Pacific) Limited (CE number AAJ321) is regulated by the Hong Kong Monetary Authority and the Securities and Futures Commission in Hong Kong and/or J.P. Morgan Broking (Hong Kong) Limited (CE number AAB027) is regulated by the Securities and Futures Commission in Hong Kong. **Korea**: This material is issued and distributed in Korea by or through J.P. Morgan Securities (Far East) Limited, Seoul Branch, which is a member of the Korea Exchange(KRX) and is regulated by the Financial Services Commission (FSC) and the Financial Supervisory Service (FSS). **Australia**: J.P. Morgan Securities Australia Limited (JPMSAL) (ABN 61 003 245 234/AFS Licence No: 238066) is regulated by ASIC and is a Market, Clearing and Settlement Participant of ASX Limited and CHI-X. **Taiwan**: J.P. Morgan Securities (Taiwan) Limited is a participant of the Taiwan Stock Exchange (company-type) and regulated by the Taiwan Securities and Futures Bureau. **India:** J.P. Morgan India Private Limited (Corporate Identity Number - U67120MH1992FTC068724), having its registered office at J.P. Morgan Tower, Off. C.S.T. Road, Kalina, Santacruz - East, Mumbai – 400098, is registered with Securities and Exchange Board of India (SEBI) as a 'Research Analyst' having registration number INH000001873. J.P. Morgan India Private Limited is also registered with SEBI as a member of the National Stock Exchange of India Limited and the Bombay Stock Exchange Limited (SEBI Registration Number – INZ000239730) and as a Merchant Banker (SEBI Registration Number - MB/INM000002970). Telephone: 91-22-6157 3000, Facsimile: 91-22-6157 3990 and Website: www.jpmipl.com. For non local research reports, this material is not distributed in India by J.P. Morgan India Private Limited. **Thailand**: This material is issued and distributed in Thailand by JPMorgan Securities (Thailand) Ltd., which is a member of the Stock Exchange of Thailand and is regulated by the Ministry of Finance and the Securities and Exchange Commission and its registered address is 3rd Floor, 20 North Sathorn Road, Silom, Bangrak, Bangkok 10500. **Indonesia**: PT J.P. Morgan Sekuritas Indonesia is a member of the Indonesia Stock Exchange and is regulated by the OJK a.k.a. BAPEPAM LK. **Philippines**: J.P. Morgan Securities Philippines Inc. is a Trading Participant of the Philippine Stock Exchange and a member of the Securities Clearing Corporation of the Philippines and the Securities Investor Protection Fund. It is regulated by the Securities and Exchange Commission. **Brazil**: Banco J.P. Morgan S.A. is regulated by the Comissao de Valores Mobiliarios (CVM) and by the Central Bank of Brazil. **Mexico**: J.P. Morgan Casa de Bolsa, S.A. de C.V., J.P. Morgan Grupo Financiero is a member of the Mexican Stock Exchange and authorized to act as a broker dealer by the National Banking and Securities Exchange Commission. **Singapore:** This material is issued and distributed in Singapore by or through J.P. Morgan Securities Singapore Private Limited (JPMSS) [MCI (P) 058/04/2019 and Co. Reg. No.: 199405335R], which is a member of the Singapore Exchange Securities Trading Limited and/or JPMorgan Chase Bank, N.A., Singapore branch (JPMCB Singapore) [MCI (P) 046/09/2018], both of which are regulated by the Monetary Authority of Singapore. This material is issued and distributed in Singapore only to accredited investors, expert investors and institutional investors, as defined in Section 4A of the Securities and Futures Act, Cap. 289 (SFA). This material is not intended to be issued or distributed to any retail investors or any other investors that do not fall into the classes of "accredited investors," "expert investors" or "institutional investors," as defined under Section 4A of the SFA. Recipients of this document are to contact JPMSS or JPMCB Singapore in respect of any matters arising from, or in connection with, the document. **Japan**: JPMorgan Securities Japan Co., Ltd. and JPMorgan Chase Bank, N.A., Tokyo Branch are regulated by the Financial Services Agency in Japan. **Malaysia**: This material is issued and distributed in Malaysia by JPMorgan Securities (Malaysia) Sdn Bhd (18146-X) which is a Participating Organization of Bursa Malaysia Berhad and a holder of Capital Markets Services License issued by the Securities Commission in Malaysia. **Pakistan**: J. P.

Morgan Pakistan Broking (Pvt.) Ltd is a member of the Karachi Stock Exchange and regulated by the Securities and Exchange Commission of Pakistan. **Saudi Arabia**: J.P. Morgan Saudi Arabia Ltd. is authorized by the Capital Market Authority of the Kingdom of Saudi Arabia (CMA) to carry out dealing as an agent, arranging, advising and custody, with respect to securities business under licence number 35-07079 and its registered address is at 8th Floor, Al-Faisaliyah Tower, King Fahad Road, P.O. Box 51907, Riyadh 11553, Kingdom of Saudi Arabia. **Dubai**: JPMorgan Chase Bank, N.A., Dubai Branch is regulated by the Dubai Financial Services Authority (DFSA) and its registered address is Dubai International Financial Centre - Building 3, Level 7, PO Box 506551, Dubai, UAE. **Russia:** CB J.P. Morgan Bank International LLC is regulated by the Central Bank of Russia. **Argentina:** JPMorgan Chase Bank Sucursal Buenos Aires is regulated by Banco Central de la República Argentina ("BCRA"- Central Bank of Argentina) and Comisión Nacional de Valores ("CNV"- Argentinian Securities Commission")

Country and Region Specific Disclosures
U.K. and European Economic Area (EEA): Unless specified to the contrary, issued and approved for distribution in the U.K. and the EEA by JPMS plc. Investment research issued by JPMS plc has been prepared in accordance with JPMS plc's policies for managing conflicts of interest arising as a result of publication and distribution of investment research. Many European regulators require a firm to establish, implement and maintain such a policy. Further information about J.P. Morgan's conflict of interest policy and a description of the effective internal organisations and administrative arrangements set up for the prevention and avoidance of conflicts of interest is set out at the following link https://www.jpmorgan.com/jpmpdf/1320742677360.pdf. This report has been issued in the U.K. only to persons of a kind described in Article 19 (5), 38, 47 and 49 of the Financial Services and Markets Act 2000 (Financial Promotion) Order 2005 (all such persons being referred to as "relevant persons"). This document must not be acted on or relied on by persons who are not relevant persons. Any investment or investment activity to which this document relates is only available to relevant persons and will be engaged in only with relevant persons. In other EEA countries, the report has been issued to persons regarded as professional investors (or equivalent) in their home jurisdiction. **Australia:** This material is issued and distributed by JPMSAL in Australia to "wholesale clients" only. This material does not take into account the specific investment objectives, financial situation or particular needs of the recipient. The recipient of this material must not distribute it to any third party or outside Australia without the prior written consent of JPMSAL. For the purposes of this paragraph the term "wholesale client" has the meaning given in section 761G of the Corporations Act 2001. J.P. Morgan's research coverage universe spans listed securities across the ASX All Ordinaries index, securities listed on offshore markets, unlisted issuers and investment products which Research management deem to be relevant to the investor base from time to time. J.P. Morgan seeks to cover companies of relevance to the domestic and international investor base across all GIC sectors, as well as across a range of market capitalisation sizes. **Germany:** This material is distributed in Germany by J.P. Morgan Securities plc, Frankfurt Branch which is regulated by the Bundesanstalt für Finanzdienstleistungsaufsicht. **Korea:** This report may have been edited or contributed to from time to time by affiliates of J.P. Morgan Securities (Far East) Limited, Seoul Branch. **Singapore:** As at the date of this report, JPMSS is a designated market maker for certain structured warrants listed on the Singapore Exchange where the underlying securities may be the securities discussed in this report. Arising from its role as designated market maker for such structured warrants, JPMSS may conduct hedging activities in respect of such underlying securities and hold or have an interest in such underlying securities as a result. The updated list of structured warrants for which JPMSS acts as designated market maker may be found on the website of the Singapore Exchange Limited: http://www.sgx.com. In addition, JPMSS and/or its affiliates may also have an interest or holding in any of the securities discussed in this report – please see the Important Disclosures section above. For securities where the holding is 1% or greater, the holding may be found in the Important Disclosures section above. For all other securities mentioned in this report, JPMSS and/or its affiliates may have a holding of less than 1% in such securities and may trade them in ways different from those discussed in this report. Employees of JPMSS and/or its affiliates not involved in the preparation of this report may have investments in the securities (or derivatives of such securities) mentioned in this report and may trade them in ways different from those discussed in this report. **Taiwan**: Research relating to equity securities is issued and distributed in Taiwan by J.P. Morgan Securities (Taiwan) Limited, subject to the license scope and the applicable laws and the regulations in Taiwan. According to Paragraph 2, Article 7-1 of Operational Regulations Governing Securities Firms Recommending Trades in Securities to Customers (as amended or supplemented) and/or other applicable laws or regulations, please note that the recipient of this material is not permitted to engage in any activities in connection with the material which may give rise to conflicts of interests, unless otherwise disclosed in the "Important Disclosures" in this material. **India:** For private circulation only, not for sale. **Pakistan:** For private circulation only, not for sale. **New Zealand:** This material is issued and distributed by JPMSAL in New Zealand only to "wholesale clients" (as defined in the Financial Advisers Act 2008). The recipient of this material must not distribute it to any third party or outside New Zealand without the prior written consent of JPMSAL. **Canada:** This report is distributed in Canada by or on behalf of J.P.Morgan Securities Canada Inc. The information contained herein is not, and under no circumstances is to be construed as an offer to sell securities described herein, or solicitation of an offer to buy securities described herein, in Canada or any province or territory thereof. The information contained herein is under no circumstances to be construed as investment advice in any province or territory of Canada and is not tailored to the needs of the recipient. **Dubai:** This report has been issued to persons regarded as professional clients as defined under the DFSA rules. **Brazil:** Ombudsman J.P. Morgan: 0800-7700847 / ouvidoria.jp.morgan@jpmorgan.com.

General: Additional information is available upon request. Information has been obtained from sources believed to be reliable but JPMorgan Chase & Co. or its affiliates and/or subsidiaries (collectively J.P. Morgan) do not warrant its completeness or accuracy except with respect to any disclosures relative to JPMS and/or its affiliates and the analyst's involvement with the issuer that is the subject of the research. All pricing is indicative as of the close of market for the securities discussed, unless otherwise stated. Opinions and estimates constitute our judgment as of the date of this material and are subject to change without notice. Past performance is not indicative of future results. This material is not intended as an offer or solicitation for the purchase or sale of any financial instrument. The opinions and recommendations herein do not take into account individual client circumstances, objectives, or needs and are not intended as recommendations of particular securities, financial instruments or strategies to particular clients. The recipient of this report must make its own independent decisions regarding any securities or financial instruments mentioned herein. JPMS distributes in the U.S. research published by non-U.S. affiliates and accepts responsibility for its contents. Periodic updates may be provided on companies/industries based on company specific developments or announcements, market conditions or any other publicly available information. Clients should contact analysts and execute transactions through a J.P. Morgan subsidiary or affiliate in their home jurisdiction unless governing law permits otherwise.

"Other Disclosures" last revised July 20, 2019.