W20: Deep Learning with Python

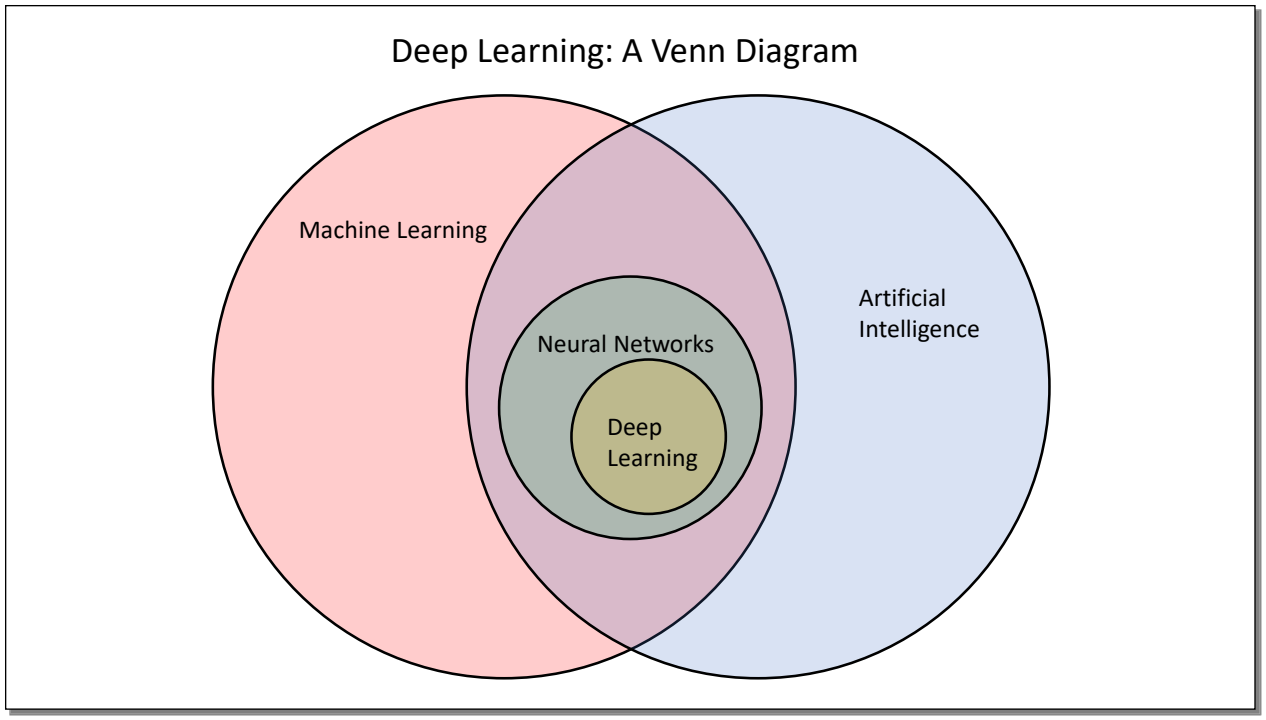
Saturday January 12, 2019, 1:00 - 5:00 pm

AAPT Winter Meeting, Houston, Texas

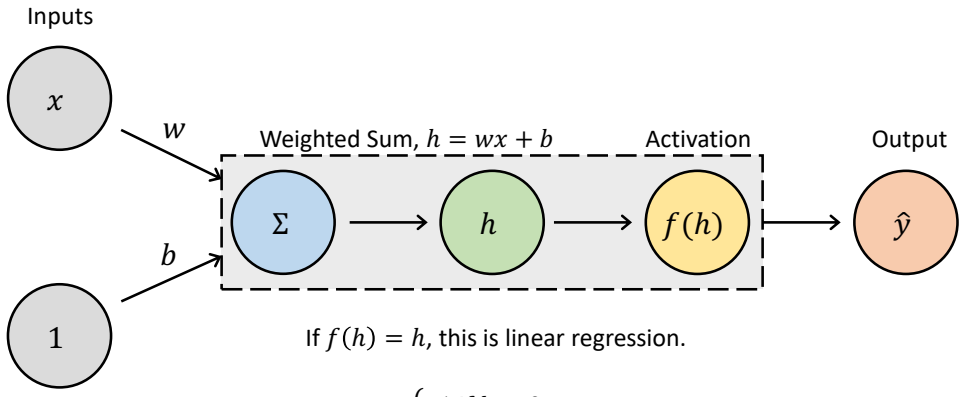
Jeff Groff, Shepherd University

Workshop Materials:

https://github.com/prof-groff/deep-learning


Deep Learning: A Venn Diagram
Machine Learning
Artificial Intelligence
Neural Networks
Deep Learning

Single-Layer, Single-Node Neural Network



If $f(h) = h$, this is linear regression.

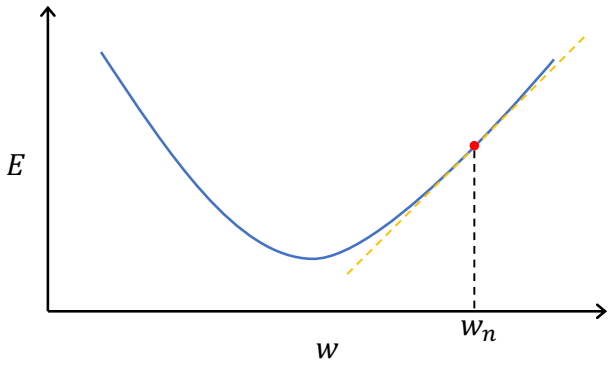
If $f(h) = \begin{cases} 1 \text{ if } h > 0 \\ 0 \text{ otherwise} \end{cases}$, this is a perceptron.

If $f(h) = \frac{1}{1+e^{-h}}$, this is logistic regression.

Gradient Descent

Goal: Find network weights that minimize the error between the model outputs (predictions) and the actual data (targets).

To minimize $E(w,b) = \frac{1}{2}(y - \hat{y}(w,b))^2$ we can iterate each weight at each time step by a value proportional to the partial derivative of the error with respect to the weight.



$$\Delta w = w_{n+1} - w_n = -\eta \frac{\partial E}{\partial w}$$

$$\Delta b = b_{n+1} - b_n = -\eta \frac{\partial E}{\partial b}$$

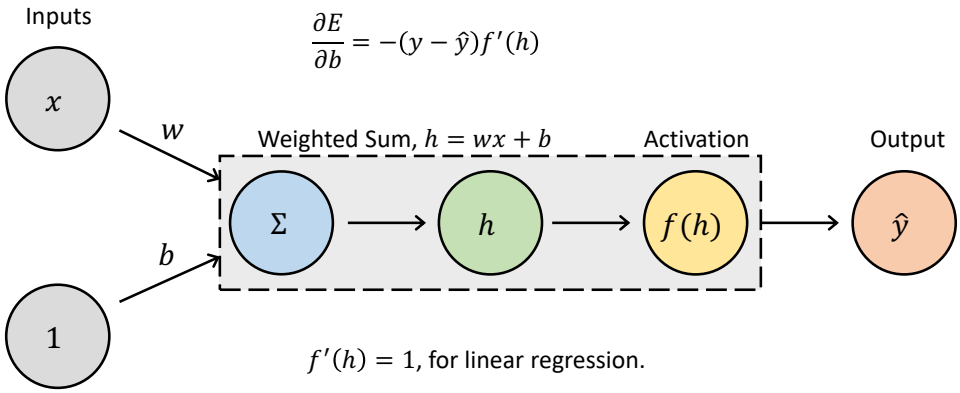
$$\Delta w = \frac{-\eta \sum_{\text{all data}} \frac{\partial E}{\partial w}}{\text{number of data records}}$$

Gradient Descent

$$E = \frac{1}{2}(y - \hat{y})^2$$

$$\frac{\partial E}{\partial w} = -(y - \hat{y})\frac{\partial \hat{y}}{\partial w} = -(y - \hat{y})f'(h)\frac{\partial h}{\partial w} = -(y - \hat{y})f'(h)x$$

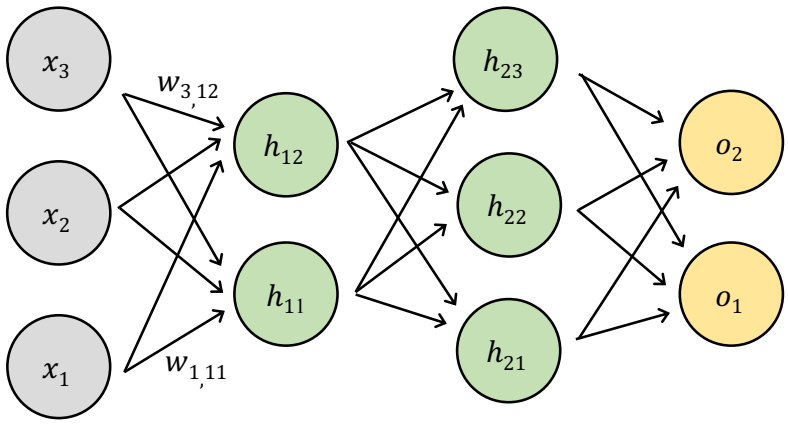
$$\frac{\partial E}{\partial b} = -(y - \hat{y})f'(h)$$

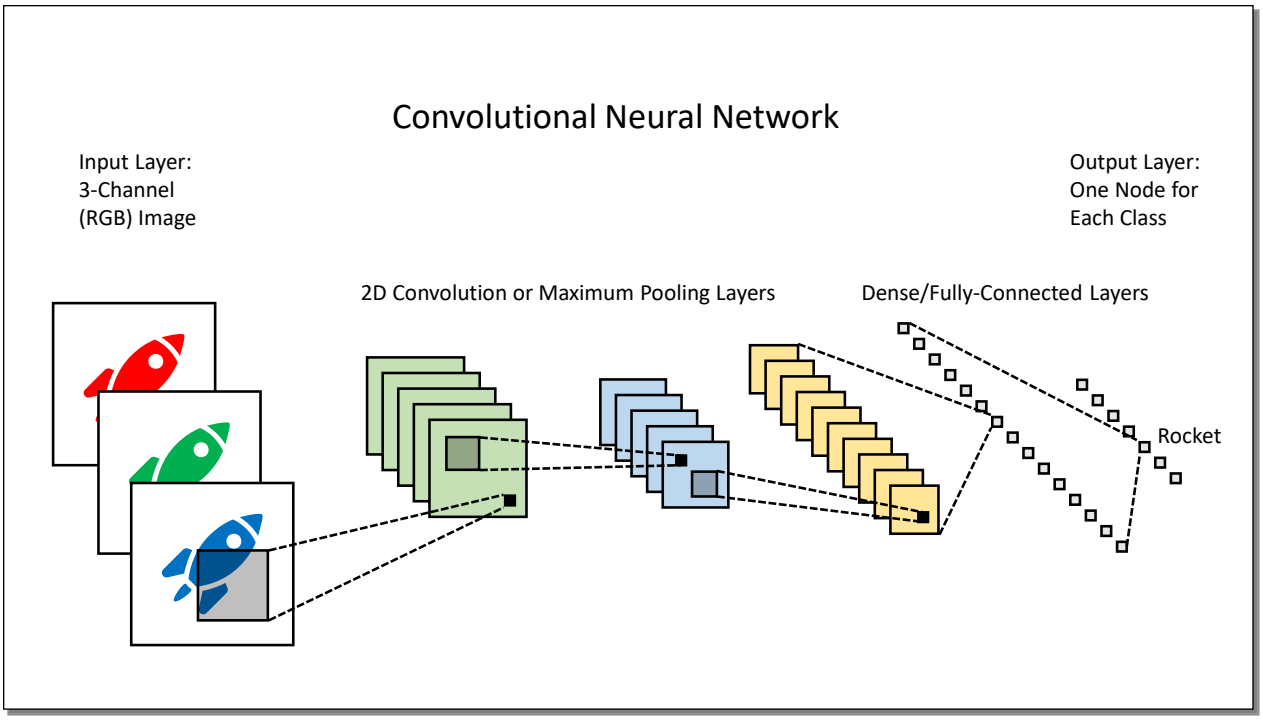


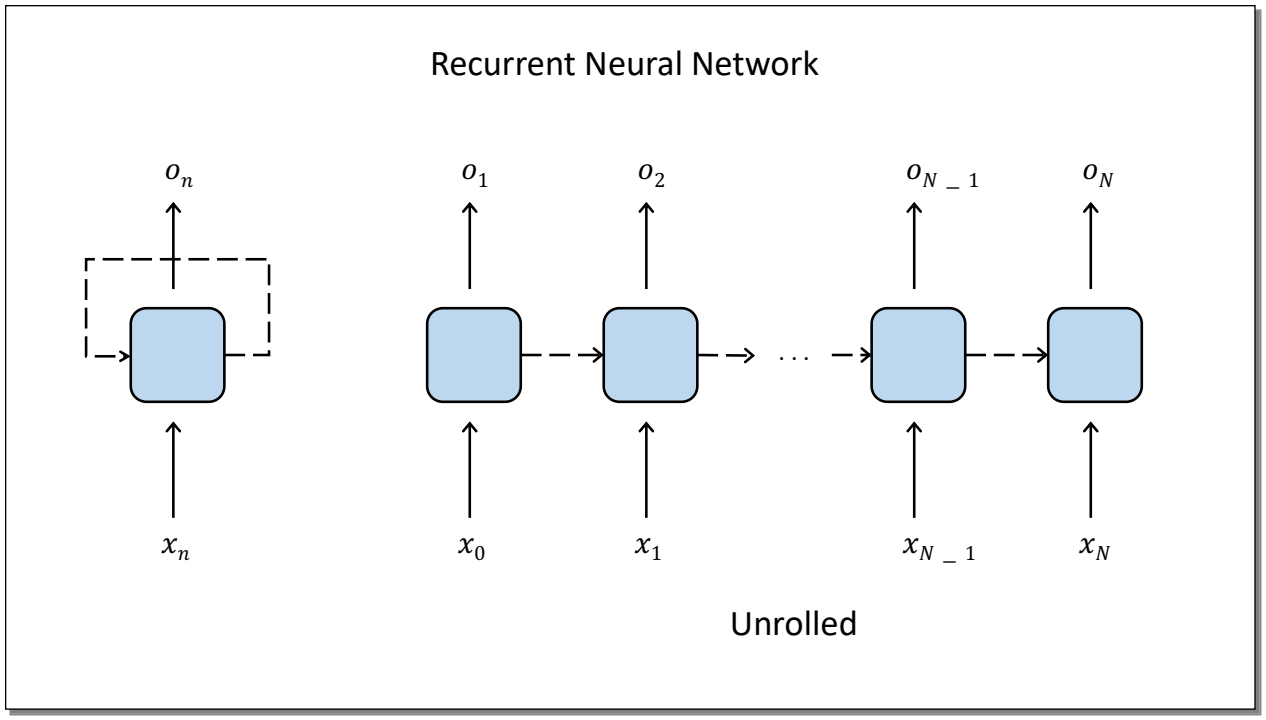
$f'(h) = 1$, for linear regression.

$f'(h) = f(h)(1 - f(h))$, for logistic regression.

Multi-Node, Multi-Layer Neural Network
Deep Learning Models Have Many Layers




Convolutional Neural Network
Input Layer:
3-Channel
(RGB) Image
2D Convolution or Maximum Pooling Layers
Dense/Fully-Connected Layers
Output Layer:
One Node for
Each Class
Rocket


Recurrent Neural Network
o_n
o_1
o_2
o_{N_1}
o_N
x_n
x_0
x_1
x_{N_1}
x_N
Unrolled

Glossary

Artificial Intelligence

Application of computer and data science toward the creation of machines that learn, perform, and respond like humans.

Machine Learning

Models and algorithms that allow machines to automatically learn pattern in data without a human operator having to teach these patterns to the machine.

Artificial Neural Network (ANN)

A computational model consisting of nodes, connections between these nodes, and rules regarding how information flows through the nodes and is processed by the nodes. Such models are reminiscent of biological neural networks where the nodes are like neurons, the connections are like synapses, and the rules governing information processing are the host of linear and non-linear responses neurons exhibit to stimuli.

Perceptron

A single-layer, single node neural network that operates as a binary classifier assigning data to one of two groups.

Deep Learning

Artificial neural networks having several to many layers. Research and experience have shown that deep networks (more layers each with fewer nodes) often perform better than wide networks (fewer layers but with more nodes in each layer). Deep learning is also inspired by the human brain where layered structures of neurons perform complex hierarchical processing. For example, the visual cortex has at least six distinct layers that each performing progressively more complicated processing.

Node

The individual elements of an ANN reminiscent of biological neurons.

Weights

The strengths assigned to connections between nodes in an ANN reminiscent of synaptic strength in biological neural networks.

Graph

A representation of the structure of an ANN specifying the nodes, layers, and connections between nodes.

Gradient Descent

A numerical algorithm for optimizing the parameters (weights) of an ANN. Stochastic gradient descent is a variation of this method that relies on randomly sampling the data at each optimization step instead of processing the entire data set. The data processed at each step is called a batch and multiple batches are processed as part of each pass through the entire data set, which is called an epoch.

Back Propagation

The process of applying gradient descent on a multi-layered ANN by propagating the errors back through the network from outputs to inputs.

Activation Function

A function applied to output of a node in an ANN. These functions allow non-linearity to be introduced into the ANN.

Hyperparameters

The parameters of an ANN that are tuned by the user, and not optimized by part of the gradient descent algorithm.

Training

The process of optimizing ANN model parameters via gradient descent. The data is often divided into three sets, one for training, one for validation during the training process, and one for testing after training. It is important to segregate the testing data from the training data to accurately assess the model performance.

Overfitting

The phenomenon of a ANN memorizing the training data so that it exhibits good accuracy on this data but poor accuracy on testing data. In other words, the model fails to generalize to new data.

Deep Learning Resources

AWS

http://aws.amazon.com

Amazon Web Services via their Elastic Compute Cloud (EC2) allows users to create computer instances with both CPUs and GPUs attached to them. Free trial periods or promotional credits may be available for new users. Also, the is a free tier of services.

Udacity

http://www.udacity.com

Udacity is an online training platform for data science, machine learning, software development, and much more. Many classes or modules are available for free.

Kaggle

http://www.kaggle.com

Kaggle is an online community of data scientists. Users can create and share online Jupyter Notebook or Python scripts and can run them in the cloud. Community members can also participate in data science competitions.

FloydHub

http://www.floydhub.com

FloydHub is a cloud-based platform for building, training, and deploying deep learning models. There is a free tier with limited capacity.

Google Cloud

http://cloud.google.com

Google Cloud allows you to set up and run computer instances in a similar fashion to AWS. There may be a free trial period and promotional credits for new users, and like AWS there is a free tier. But, there is no educator program like what AWS offers.

Microsoft Azure Notebooks

http://notebooks.azure.com

Microsoft's Azure service also allows user to create cloud compute instances. But, Azure also offers free online Jupyter Notebooks. This is a great service but has been somewhat resource constrained in the past.

Keras Examples

https://github.com/keras-team/keras/tree/master/examples

Python scripts showing examples of various deep learning models implemented in Keras including vision models, text models, and generative models.

Online Learning Resources

The Unreasonable Effectiveness of Recurrent Neural Network
Andrej Karpathy
http://karpathy.github.io/2015/05/21/rnn-effectiveness

Deep Learning: An MIT Press Book
Ian Goodfellow, Yoshua Bengio, and Aaron Courville
http://www.deeplearningbook.org

Neural Network and Deep Learning
Michael Nielson
http://neuralnetworksanddeeplearning.com/index.html

How to Build a Simple Image Recognition System with TensorFlow (Part 1)
Wolfgang Beyer
https://www.wolfib.com/Image-Recognition-Intro-Part-1

Deep Learning in the Cloud - Part 1

AWS DLAMI (Deep Learning Amazon Machine Image) - Set Up Instructions

1) Create an AWS Account

Go to https://aws.amazon.com to sign up. This account can be linked with an existing Amazon account.

2) Optional: Join AWS Educate

Go to https://aws.amazon.com/education/awseducate to sign up for an educator account to earn free credits to apply to your AWS account. During the application process you must specify a class you plan to use AWS in. If you don't have a specific class in mind, then pick a class for which you could explore the use of AWS. Once your application is approved you will get an email with a promo code that you can enter into your AWS billing console for a $40 credit. Note that the email address you use to sign up for AWS Educate must be your official school email address, but you can use a different email address with AWS.

3) Request Use of a GPU

Log into Amazon Web Services at https://aws.amazon.com and navigate to Services > Compute > EC2 > Limits and request an increase in instance limits to allow you to use an instance with a GPU attached to it. Two recommended instance classes are g3.4xlarge (currently $1.1/hr) or p3.2xlarge (much more powerful GPU but $3/hr). It may take a day for your request to be approved.

4) Create a New Instance

Log into AWS at https://aws.amazon.com and navigate to Services > Compute > EC2 and click on the Launch Instance button. Select the AMS Marketplace option and search for Deep Learning AMI (or DLAMI). Select one of the precompiled images, which include common deep learning toolset such as NVIDIA CUDA, cuDNN, python, TensorFlow, Keras, etc. I use the Deep Learning AMI (Ubuntu) image. As you progress through the image creation process there are many options. The defaults are mostly okay. But, when you are selecting the instance type make sure to select the type requested above. Also, if you plan to use Jupyter Notebook with your instance you will need to add the following networking rule to make this possible:

type: Custom TCP Rule

protocol: TCP

port range: 8888

source: Anywhere 0.0.0.0/0, ::/0

When you launch your instance you will be instructed to either launch the instance with an existing key pair or to create a new key pair if you do not already have one. This key pair is necessary to allow you to SSH into your instance once it is running. Don't lose your copy. Once your instance is created you can start and stop it at Services > Computer > EC2 > Instances. Make sure you stop the instance when it is not in use or you will continue to be billed for its resource consumption.

Deep Learning in the Cloud - Part 2

SSH into an AWS Instance

If you are working on a Mac running OS X or a computer running Linux, login into your instance requires some shell commands. First, navigate to the directory were your copy of the key pair attached to the instance resides and change the permissions on the file.

```
chmod 400 <AWS_key_name>.pem
```

Then, SSH into the instance using its public DNS name. Here the instance is running Ubuntu, so the username is ubuntu.

```
ssh -i "<AWS_key_name>.pem" ubuntu@<public_DNS_name_prefix>.amazonaws.com
```

Connecting to an Instance with PuTTY

If you are working on a Windows computer download and install the SSH client called PuTTY.

https://www.putty.org

Converting AWS Key

Next, you must convert the *.pem key file associated with your instance into a *.ppk format that PuTTY understands using PuTTYgen, a tool installed with PuTTY. Open PuTTYgen and select RSA as the type of key to generate. Then select `Load` and navigate to your *.pem file. You might have to change the file extension filter to show files with all extensions. Open your *.pem file and click `Save private key` and save it as a *.ppk file dismissing warnings about saving the key without a passphrase (or add a passphrase if you like but this makes the login process more cumbersome).

Setting up PuTTY

With your converted key in hand, open PuTTY and open the Category > Sessions window. Enter your username and the instance's DNS name in the hosts box like follows. Here the instance is running Ubuntu, so the username is ubuntu.

```
ubuntu@<public_DNS_name_prefix>.amazonaws.com
```

Select the radio button for `SSH` under `Connection type:` and ensure that the `Port` is set to 22. Navigate to the Category > Connection > SSH > Auth window and use the browse button under `Private key file for authorization` to find your *.ppk file created above.

Navigate to the Category > Connection > SSH > Tunnels window and set up port forwarding on port 8888 to allow Jupyter Notebook connections. In the `Source port` box enter 8888 and in the `Destination` box enter 127.0.0.1:8888 then click the `Add` button.

To save the configuration for future use navigate back to Category > Sessions and enter a name in the `Saved Sessions` box and then click the `Save` button. When returning to PuTTY later you can select this saved session and `Load` it. Click the `Open` button to launch a connection with the instance.

Deep Learning in the Cloud - Part 3

Installing and Running Jupyter Notebook

It is possible that Jupyter Notebook is not installed by default on the AWS DLAMI image. Installation is easy. Once you SSH into the instance using a terminal or PuTTY install it for both python2 and python3 with the following commands:

`pip install ipykernel jupyter notebook` `pip3 install ipykernel jupyter notebook`
The `ipykernel` package allows you to change between python2 and python3 kernels inside of jupyter notebook. To launch Jupyter Notebook type the following:

```
jupyter notebook
```

A URL with a token string will be displayed in the terminal. Copy (simply highlighting achieves this) and paste this URL and token into a browser window. If port forwarding is set up correctly (instruction for doing this in PuTTY are given in `PuTTY-login.md`) Jupyter Notebook running on the remote instance will load in your browser. To do port forwarding on Mac OSX run the following command in the terminal.

```
ssh -i <AWS_key_name>.pem -N -f -L 8888:localhost:8888
ubuntu@<public_DNS_name_prefix>.amazonaws.com
```