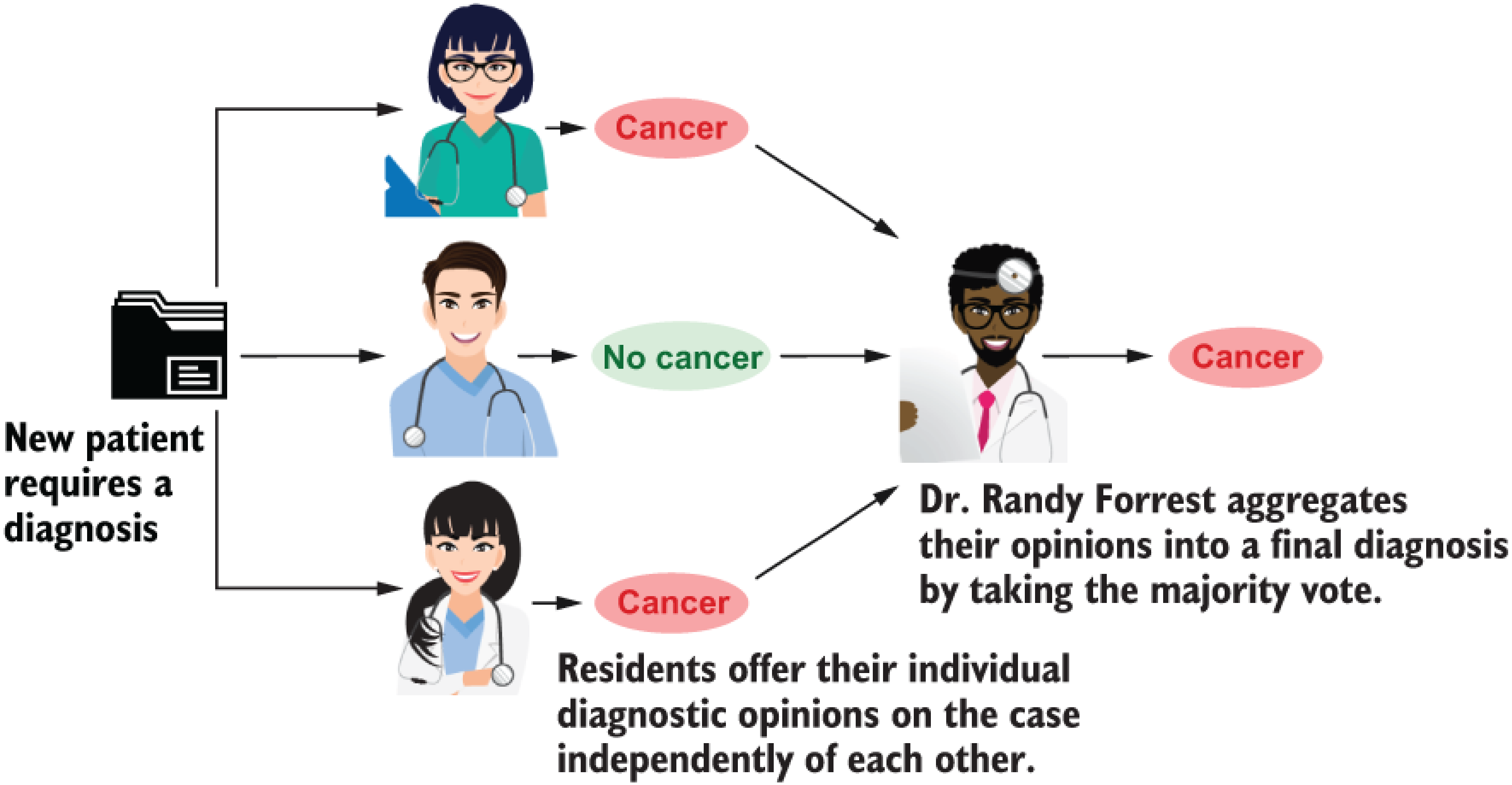
Ensembles

Dr. Mongkol Ekpanyapong

What is Ensembles?



New patient requires a diagnosis

Cancer

No cancer

Cancer

Residents offer their individual diagnostic opinions on the case independently of each other.

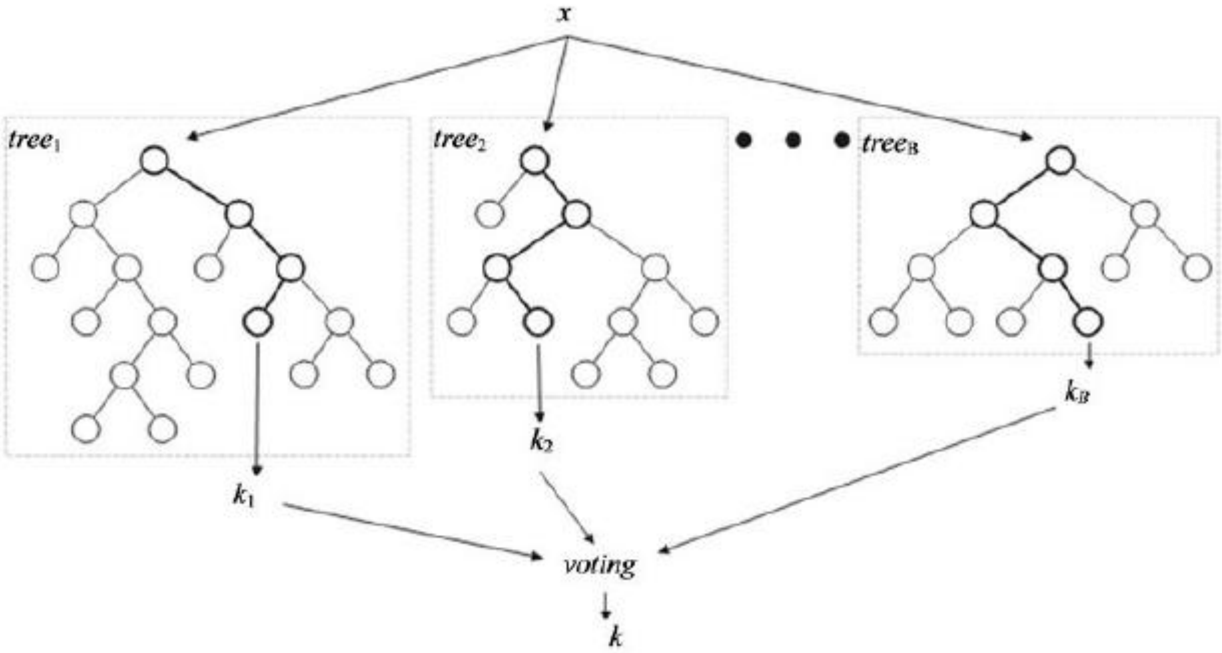
Cancer

Dr. Randy Forrest aggregates their opinions into a final diagnosis by taking the majority vote.

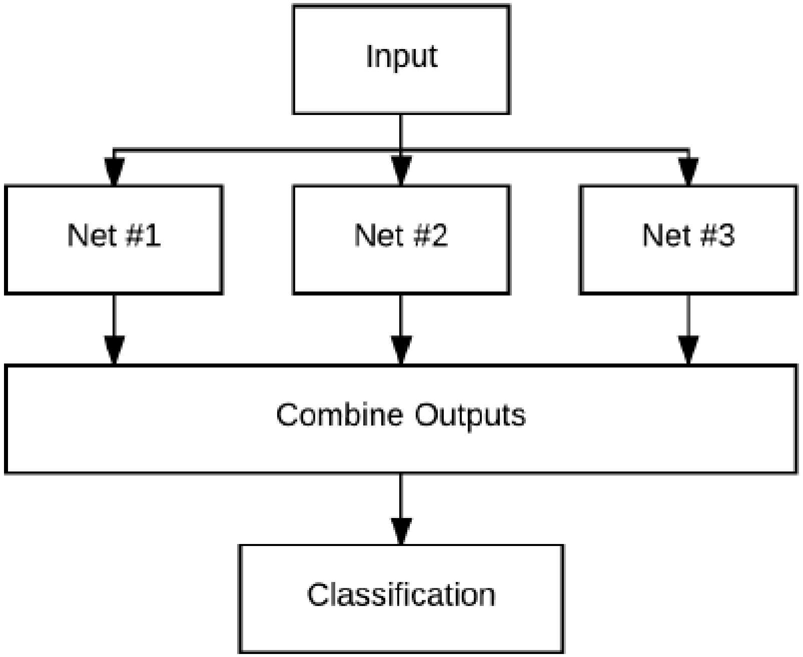
Ensembles

- Ensemble methods are the process of taking multiple classifiers and aggregating them into one big meta-classifier
- In Machine Learning, methods such as AdaBoost, and Random Forests are Ensemble methods

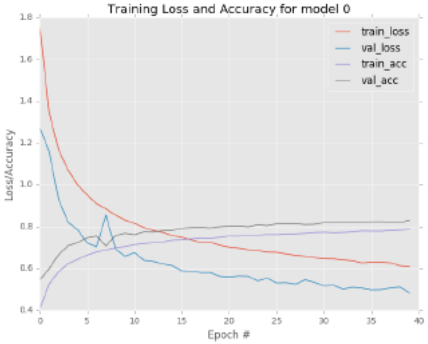
Random Forests



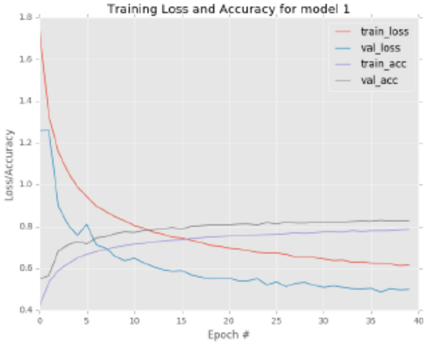
Multiple Network



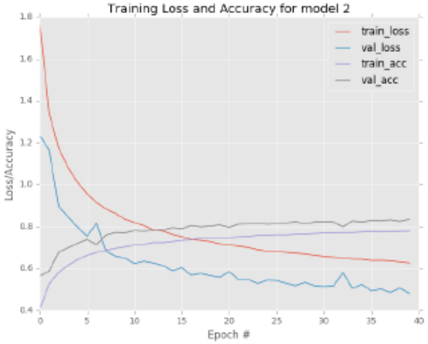
Accuracy



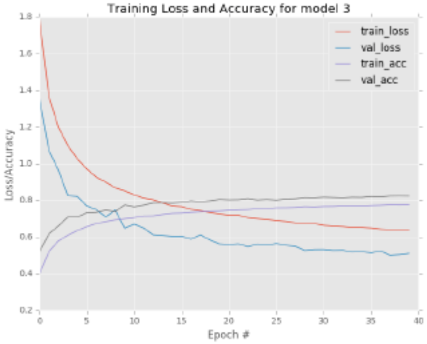
Training Loss and Accuracy for model 0



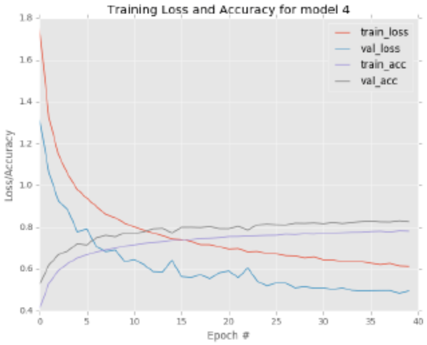
Training Loss and Accuracy for model 1



Training Loss and Accuracy for model 2



Training Loss and Accuracy for model 3



Training Loss and Accuracy for model 4

Training Multiple Networks

```
from keras import layers
from keras import models
from keras.datasets import cifar10
from sklearn.preprocessing import LabelBinarizer
import numpy as np

model1 = models.Sequential()
model1.add(layers.Conv2D(64, (5, 5), activation='relu', input_shape=(32, 32, 3)))
model1.add(layers.MaxPooling2D((2, 2)))
model1.add(layers.Conv2D(64, (5, 5), activation='relu'))
model1.add(layers.MaxPooling2D((2, 2)))
model1.add(layers.Flatten())
model1.add(layers.Dense(10, activation='softmax'))
model1.summary()
```

```
model2 = models.Sequential()
model2.add(layers.Conv2D(96, (5, 5), activation='relu', input_shape=(32, 32, 3)))
model2.add(layers.MaxPooling2D((2, 2)))
model2.add(layers.Conv2D(32, (5, 5), activation='relu'))
model2.add(layers.MaxPooling2D((2, 2)))
model2.add(layers.Flatten())
model2.add(layers.Dense(10, activation='softmax'))
model2.summary()

model3 = models.Sequential()
model3.add(layers.Conv2D(32, (5, 5), activation='relu', input_shape=(32, 32, 3)))
model3.add(layers.MaxPooling2D((2, 2)))
model3.add(layers.Conv2D(96, (5, 5), activation='relu'))
model3.add(layers.MaxPooling2D((2, 2)))
model3.add(layers.Flatten())
model3.add(layers.Dense(10, activation='softmax'))
model3.summary()
```

```
print("[INFO] loading CIFAR-10 data...")
((trainX, trainY), (testX, testY)) = cifar10.load_data()
trainX = trainX.astype("float") / 255.0
testX = testX.astype("float") / 255.0
lb = LabelBinarizer()
trainY = lb.fit_transform(trainY)
testY = lb.transform(testY)
labelNames = ["airplane", "automobile", "bird", "cat", "deer",
              "dog", "frog", "horse", "ship", "truck"]

model1.compile(optimizer='Adam',
             loss='categorical_crossentropy',
             metrics=['accuracy'])
```

```
H1 = model1.fit(trainX, trainY, validation_data=(testX, testY),
                batch_size=250, epochs=10, verbose=1)
model2.compile(optimizer='Adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
H2 = model2.fit(trainX, trainY, validation_data=(testX, testY),
                batch_size=250, epochs=10, verbose=1)
model3.compile(optimizer='Adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
H3 = model3.fit(trainX, trainY, validation_data=(testX, testY),
                batch_size=250, epochs=10, verbose=1)
predictions = []
predictions.append(model1.predict(testX,batch_size=64))
predictions.append(model2.predict(testX,batch_size=64))
predictions.append(model3.predict(testX,batch_size=64))

predictions = np.average(predictions, axis=0)
from sklearn.metrics import classification_report
print(classification_report(testY.argmax(axis=1),
                predictions.argmax(axis=1), target_names=labelNames))
```

Homework

- Design a majority vote system of 3 simple CNN model of your choice. Comparing it with the result of a single CNN model on MNIST Cloth dataset.

Questions?