CQNN: Convolutional Quadratic Neural Networks

Pranav Mantini and Shishir K. Shah
Department of Computer Science
University of Houston

Image Classification

- Image classification is a fundamental task in computer vision.
- Categorize images by visual content.

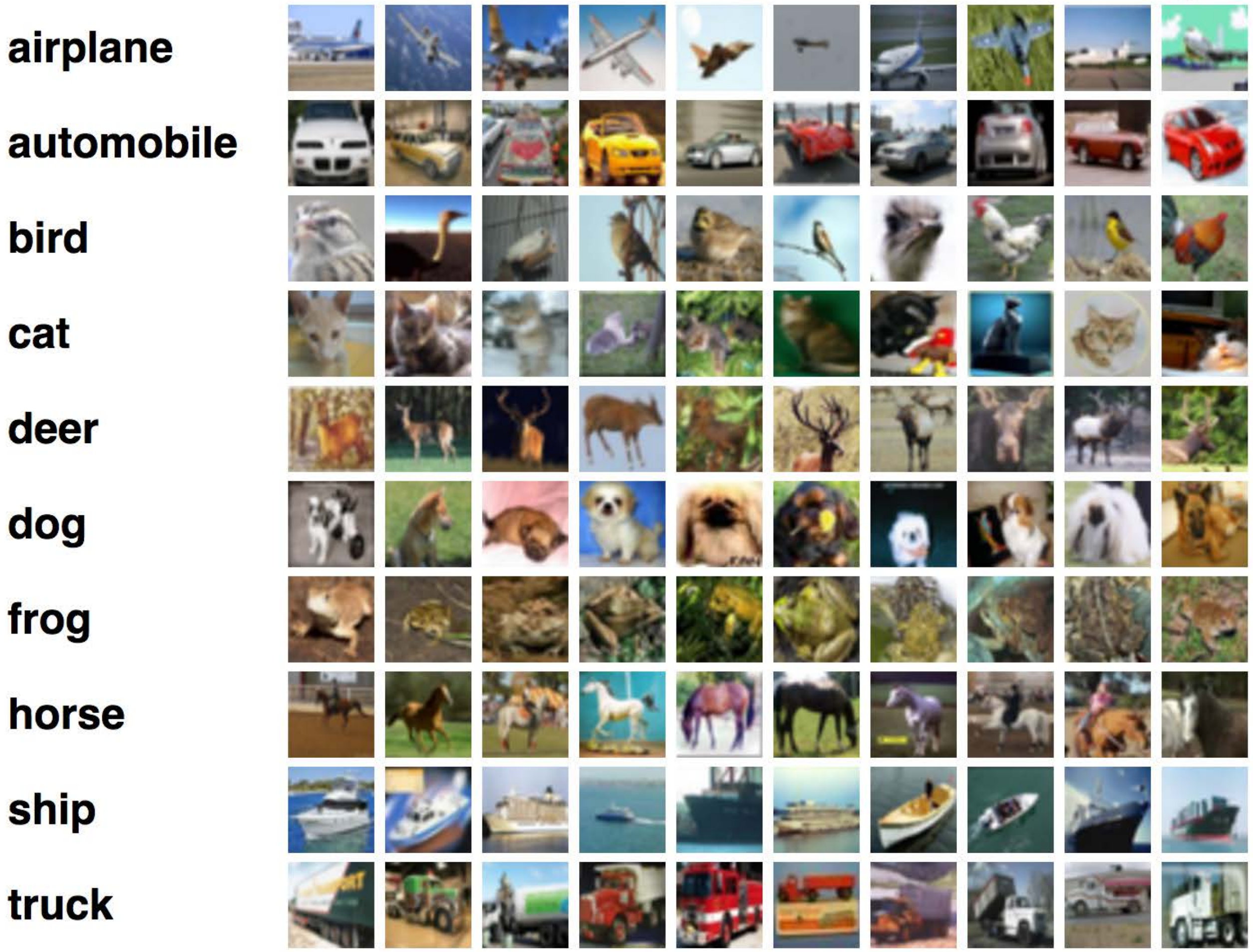


Image source: https://www.cs.toronto.edu/~kriz/cifar.html

Overall Framework

Input Image

Feature Extractions

Bow, HOG, HAAR,
LBP

Features

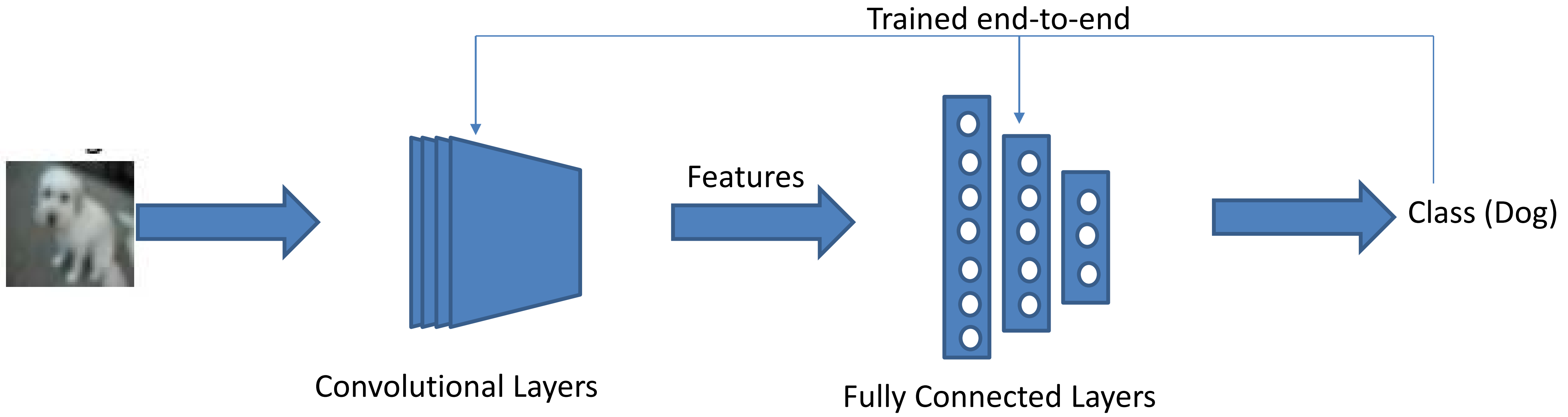
Classification

Regression, SVM,
K-Means, Naïve Bayes

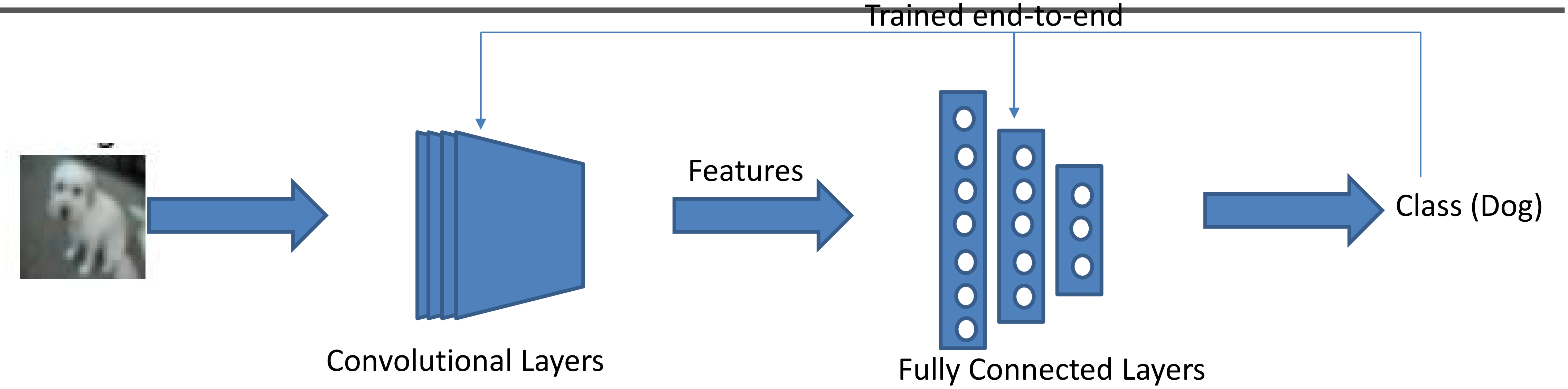
Class (Dog)

Convolutional Neural Network (CNN)

- CNN architecture has proven to be an efficient solution.



Related Work



Neocognitron	Spatially correlated sequence (patches) of image pixel, apply linear function	Down-sampling layers	Fukushima et al.
LeNet-5	Stacking convolution layers followed by pooling	Fully connected layers	Lecun et al.
AlexNet	Use of overlapping pooling and Rectified Linear Unit (ReLU) activation.	-	Krizhevsky et al.

Related Work

- Modular structure that allowed for scaling in multiple dimensions.

Visual Geometry Group (VGG) VGG-11, VGG-16, and VGG-19	Vertical scaling
GoogLeNet (Szegedy *et al.*)	Horizontally scaling
Residual networks (ResNet) (He *et al.*)	Introduced skipped connections
DenseNet (Huang *et al.*)	Extended skipped connections
Network in Network (NIN) architecture (Lin *et al.*)	Introducing non-linearity using micro-neural networks
Other research contributions	Dropout layers, batch normalization, data augmentation techniques, and etc

Related Work

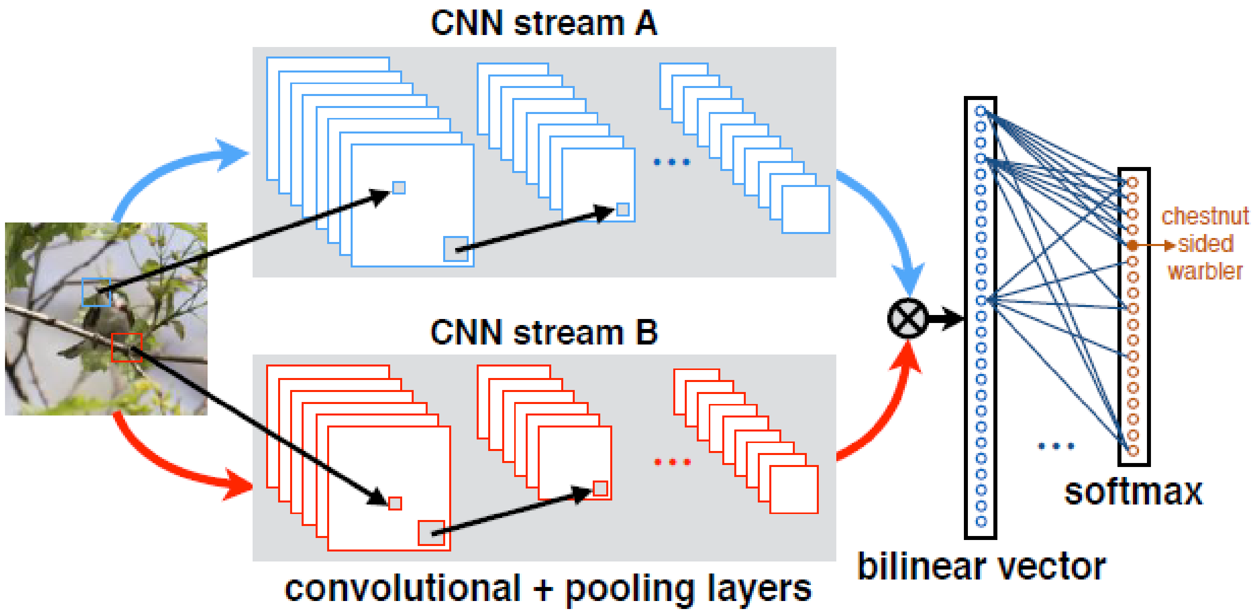
- Modular structure that allowed for scaling in multiple dimensions.

Visual Geometry Group (VGG) VGG-11, VGG-16, and VGG-19	Vertical scaling
GoogLeNet (Szegedy *et al.*)	Horizontally scaling
Residual networks (ResNet) (He *et al.*)	Introduced skipped connections
DenseNet (Huang *et al.*)	Extended skipped connections
Network in Network (NIN) architecture (Lin *et al.*)	Introducing non-linearity using micro-neural networks
Other research contributions	Dropout layers, batch normalization, data augmentation techniques, and etc

The atomic operation for these models has remained a **linear unit** (neuron)
Scale the Order of the Neuron: Hypothesizing the atomic operation to be performed by a **quadratic unit**.

Bilinear CNN's

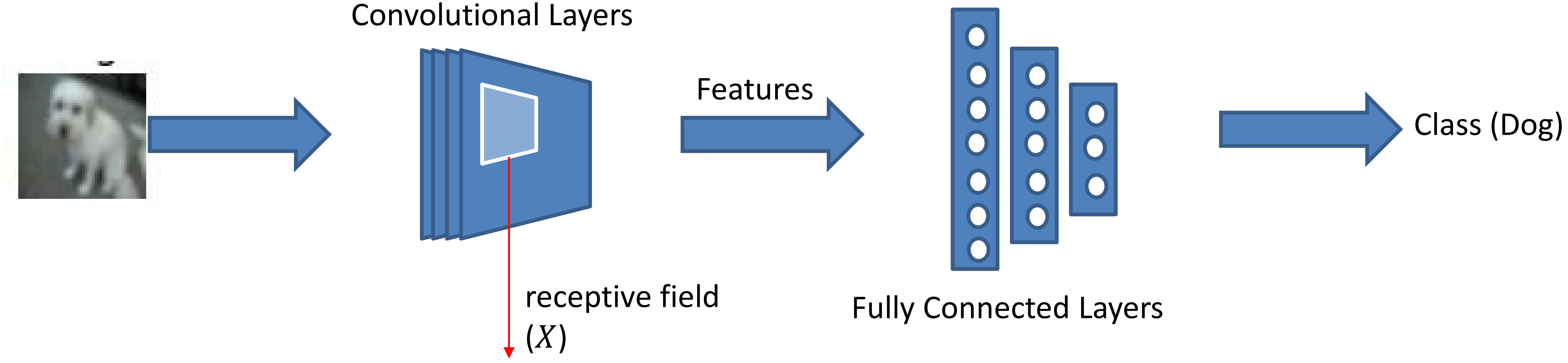
- Closely related to our work.
- Extracts features from two linear CNN networks,
- Performs an outer product to create bilinear features



Higher Order Neural Networks

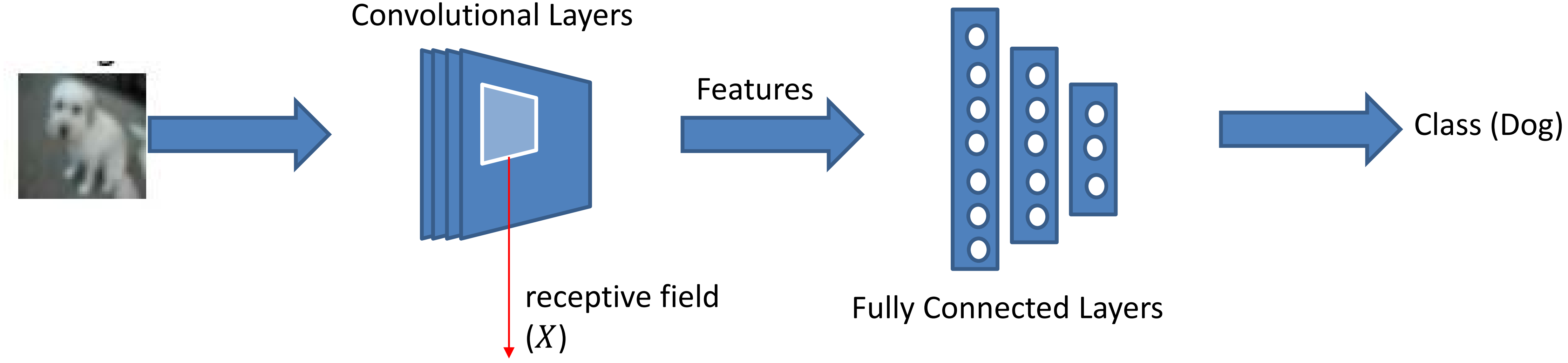
Name	Neural Function
Linear	$WX + b$
Sigma Pi Networks (Giles and Maxwell)	$WX + \boldsymbol{W'} \sum \boldsymbol{X_i X_j} + b$
Pi-Sigma network	$\prod (W_i X_i + b_i)$
Quadratic (Milenkovic *et al.*)	$WX + \boldsymbol{W'X^2} + b$ and $WX + \boldsymbol{W'X^2} + \boldsymbol{W'} \sum \boldsymbol{X_i X_j} + b$
Quadratic Junctions (DeClaris and Su)	$(WX + b)(WX + b) + b'$
Quadratic (Fan *et al.*)	$(\boldsymbol{W_1}X + \boldsymbol{b_1})(\boldsymbol{W_2}X + \boldsymbol{b_2}) + \boldsymbol{W'X^2} + b'$

CQNN: Convolutional Quadratic Neural Networks



CNN $\quad f(X) = WX + b \quad\quad f(X) = WX + b$

CQNN: Convolutional Quadratic Neural Networks



CNN

$$f(X) = WX + b \qquad f(X) = WX + b$$

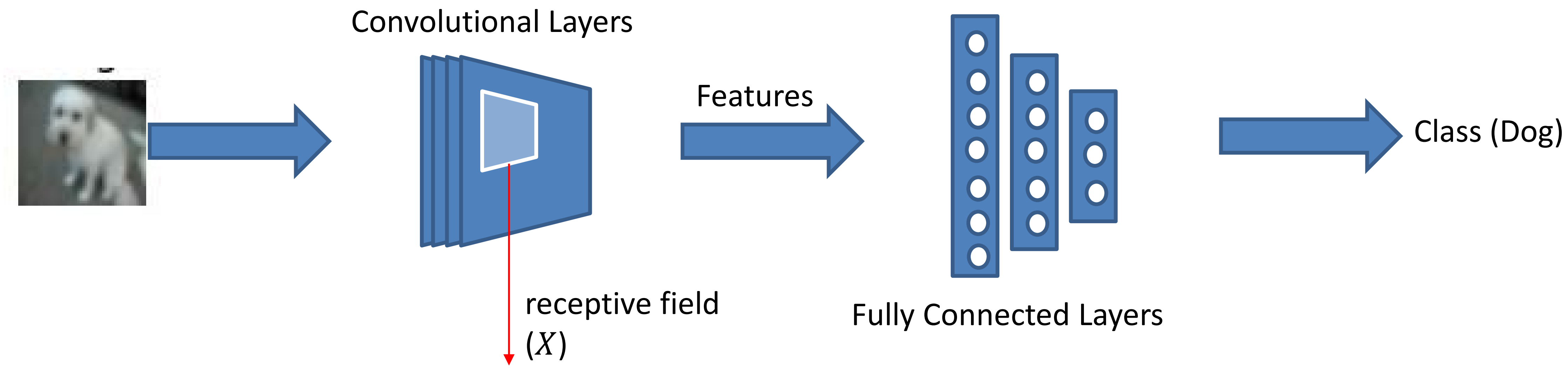
CQNN

$$q(X) = X'^T W_q X'$$

$$X'^T = \{X^T|1\} = \{x_1, x_2, ..., x_d, 1\}$$

$$W_q = \begin{bmatrix} w'_{1,1} & w'_{1,2} & \cdot & \cdot & w'_{1,d+1} \\ w'_{2,1} & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ w'_{d,1} & \cdot & \cdot & \cdot & w'_{d+1,d+1} \end{bmatrix}$$

CQNN: Convolutional Quadratic Neural Networks



CNN	$f(X) = WX + b$	$f(X) = WX + b$
CQNN	$q(X) = X'^T W_q X'$	$f(X) = WX + b$

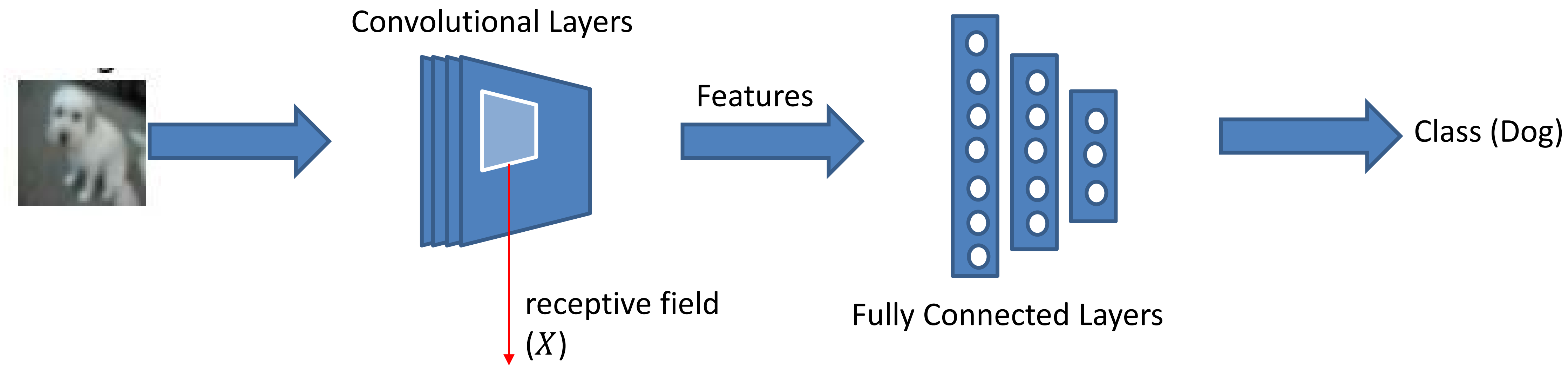
Exponential Increase in Parameters

Logical to pursue research toward building **higher order neural networks**
Studies have largely been abandoned due to the **exponential growth in parameters** with input dimensions.

Input dimensions (**d**)

Scenario	CNN	CQNN
1 Neuron	$(d+1)$	$(d+1)^2$
l layers $\{n_1, n_2, \dots, n_l\}$ neuron	$(d+1)n_1 + \Sigma_{i=1}^{l}(n_i+1)n_{i+1}$	$(d+1)^2 n_1 + \Sigma_{i=1}^{l}(n_i+1)^2 n_{i+1}$
d = 1000, 1 layer, 100 neurons	100100	100200100

CQNN: Convolutional Quadratic Neural Networks



Since kernel sizes are limited to 3X3, 5X5, …

CQNN

$$q(X) = X'^T W_q X' \qquad f(X) = WX + b$$

Motivation: Image features are extracted using higher-order functions allowing the network to learn complex representation (compared to linear functions)

Experimental Design

1. Does scaling the order of neuron improve performance?
2. Can we use ideas that have work for linear neurons on quadratic neurons?

Neural Networks

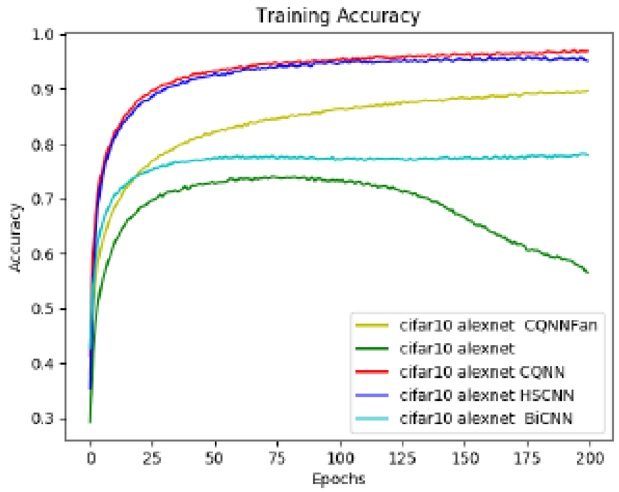
- Two base architectures: AlexNet, Resnet
- Compare performance of base networks with

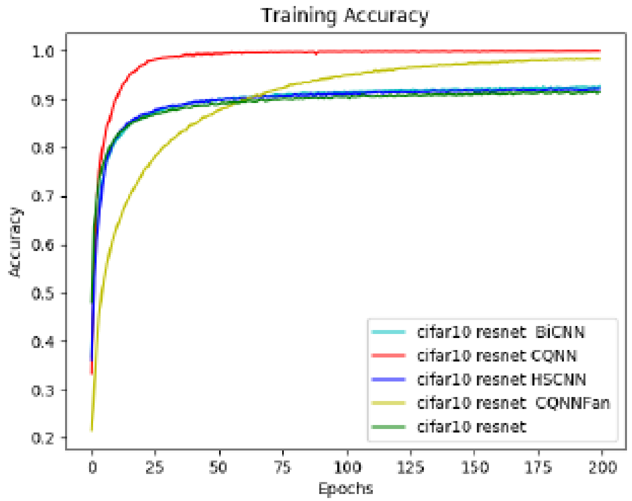
CQNN	Replace all neurons with quadratic neurons. Use exponential linear units (ELu) for activation
CQNNFan	Replace all neurons with quadratic neurons. Defined as $f(x) = (W_1 X + b_1)(W_2 X + b_2) + (W_3 X + b_3)$
Bilinear Networks	Two parallel networks to extract features, compute outer product for classification.
Horizontally Scaled Networks	Use linear neuron, but scale network horizontally. They have approximately the same number of neurons as the quadratic versions.

Datasets

- CIFAR 10 (60,000 images):
  – 10 classes with 6,000 images per class.
  – Each class has 5000 train images, and 1000 for testing.
- CIFAR 100 (60,000 images):
  – 100 classes with 600 images per class.
  – Each class has 500 are used for training and 100 for testing.

Cifar 10: Training

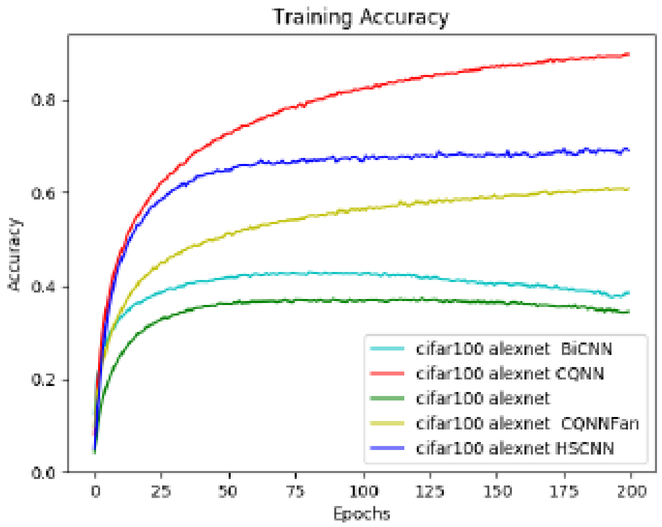


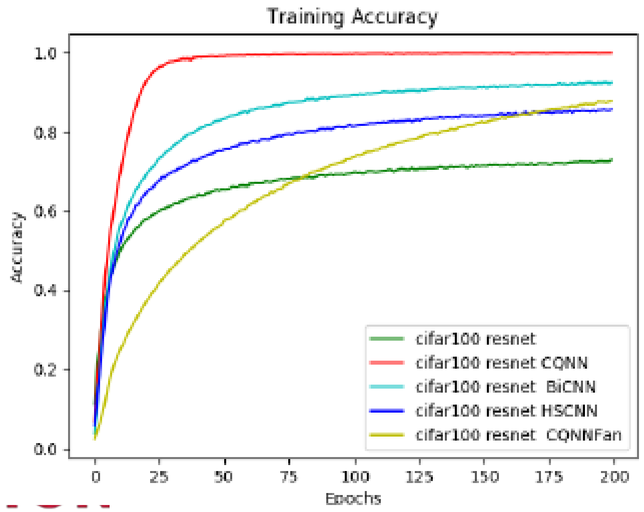


- BiCNN and CQNNFan versions of the AlexNet show improvement over the base network
- HSCNN and CQNN versions outperform the other variants.
- CQNN performs slightly better than HSCNN variant.
- HSCNN and BiCNN variants of ResNet show no improvement in the learning process.
- Quadratic version shows a clear improvement

Cifar 100: Training

- Cifar-100 is a more complex dataset than the Cifar-10.
- The effects of HSCNN and CQNNs are evident in this case.
- HSCNN and CQNN of AlexNet shows a keen improvement in the training accuracy
- The quadratic version of Resnet shows an improvement in the training accuracy.



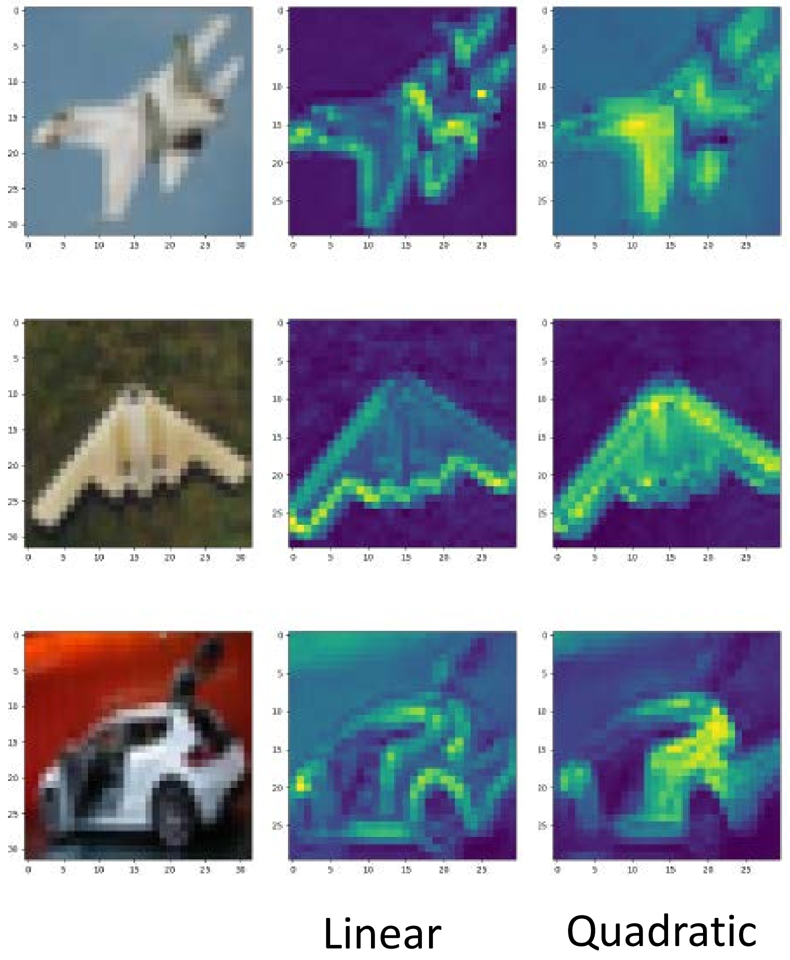


Testing Accuracy

Dataset	Model	Parameters	Accuracy	
Cifar - 10	AlexNet	1.2M	0.63	
	AlexNet HSCNN	32.2M	**0.88**	
	AlexNet BiCNN	53.2M	0.79	13%
	AlexNet CQNNFan	2.2M	0.87	24%
	AlexNet CQNN	31.4M	**0.88**	**25%**
	ResNet	.2M	0.85	
	ResNet HSCNN	130.6M	0.87	**2%**
	ResNet BiCNN	0.5M	0.84	
	ResNet CQNNFan	0.8M	0.84	
	ResNet CQNN	128M	**0.91**	**6%**
Cifar - 100	AlexNet	1.2M	0.41	
	AlexNet HSCNNs	32.2M	0.56	**15%**
	AlexNet BiCNN	64M	0.41	
	AlexNet CQNNFan	2.3M	0.58	**17%**
	AlexNet Quadratic	31.4M	**0.63**	**22%**
	ResNet	.2M	0.60	
	ResNet HSCNN	130.8M	0.59	
	ResNet BiCNN	0.9M	0.51	
	ResNet CQNNFan	0.8M	0.55	
	ResNet CQNN	128M	**0.66**	**6%**

Activations from first Convolutional Layer

- Linear convolutions layers learn to extract edges in the initial stages that are later used for classification.
- Extracting the whole object of interest and then encode its features for classification



Drawbacks

Dataset	Model	Parameters	Accuracy
Cifar - 10	AlexNet	1.2M	0.63
	AlexNet HSCNN	32.2M	**0.88**
	AlexNet BiCNN	53.2M	0.79
	AlexNet CQNNFan	2.2M	0.87
	AlexNet CQNN	31.4M	**0.88**
	ResNet	.2M	0.85
	ResNet HSCNN	130.6M	0.87
	ResNet BiCNN	0.5M	0.84
	ResNet CQNNFan	0.8M	0.84
	ResNet CQNN	128M	**0.91**
Cifar - 100	AlexNet	1.2M	0.41
	AlexNet HSCNNs	32.2M	0.56
	AlexNet BiCNN	64M	0.41
	AlexNet CQNNFan	2.3M	0.58
	AlexNet Quadratic	31.4M	**0.63**
	ResNet	.2M	0.60
	ResNet HSCNN	130.8M	0.59
	ResNet BiCNN	0.9M	0.51
	ResNet CQNNFan	0.8M	0.55
	ResNet CQNN	128M	**0.66**

AlexNet consumes **25 times** more parameters and floating-point operations (FLOPS)
ResNet20 like architecture consumes **220 times** more parameters and flops

Conclusion

- CQNNs show clear improvement in training behavior and testing accuracy compared to conventional CNN architecture.
- Concepts such as skipped connections are applicable for quadratic networks.
- However they requires too many parameters.