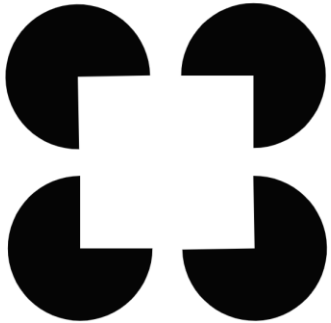
Comparison between Human Vision and Computer Vision to perceive Kanizsa Illusion.

A computational model to perceive Kanizsa illusion

Sagar B Dollin Adheena Reji Akshat Govekar

MSc Cognitive Science, IIT Delhi.

Kanizsa Illusion



Kanizsa illusion square
Image credits: Adheena Reji

- The illusory square appears brighter than the background.
- In reality the illusory triangle and the background are of same color and brightness
- Edges are perceived for the illusory triangle,but there are no edges for the illusory triangle.
- Predictive networks are responsible for illusion

Can Computer vision see such illusions?

- By computer Vision we mean popular CNN models used for computer vision applications.
- Research shows that CNN cannot perceive kanizsa illusion. (Trained on Real Square , Tested on Kanizsa square)
- In this project we evaluate the predictive coding feedback network proposed by Zhaoyang Pang et al.

Proposed method (by Zhaoyang Pang et al.)

- The model consists of 3 feedforward encoding layers, e1,e2,e3, and three corresponding decoding generative layers d0,d1,d2, whose errors are used to update the activity of the encoding layer below them at each timestamp. The error of a layer n at any timestamp t can be given as,

$$\epsilon_n = e_n - d_n.$$

- To the hierarchical encoding layer, we attach a binary classifier. That is used to classify images as either square or Pacman.
- This binary classifier acts as the Inferior Temporal Cortex in recognizing the objects in the image
- The predictive encoder model acts as a preprocessing mechanism like that of occipital lobe.

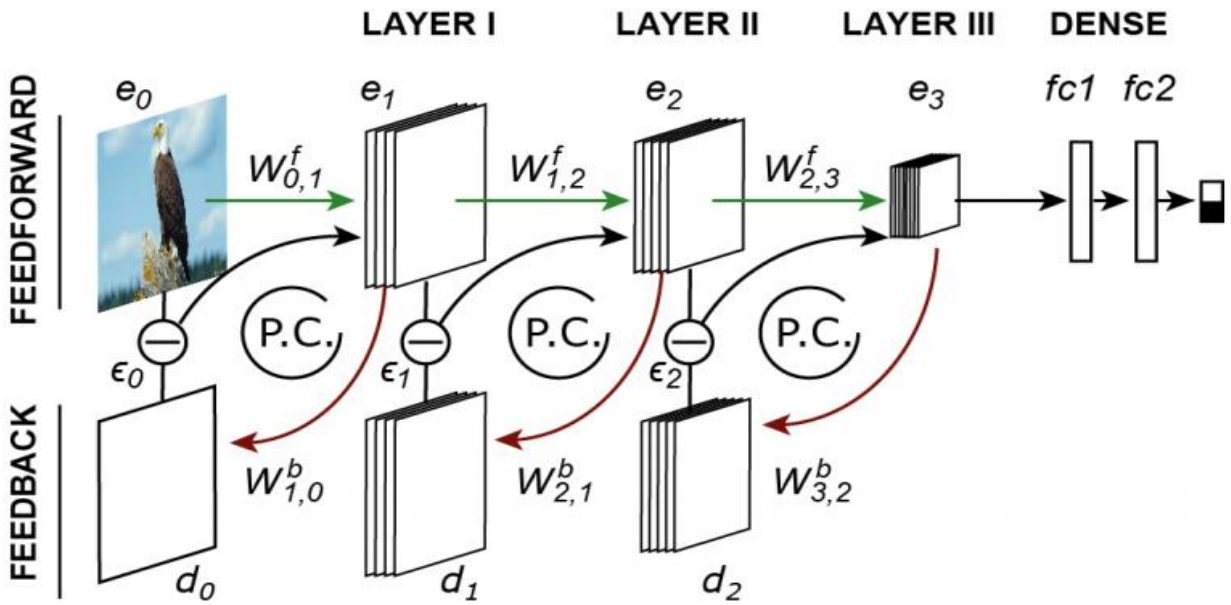
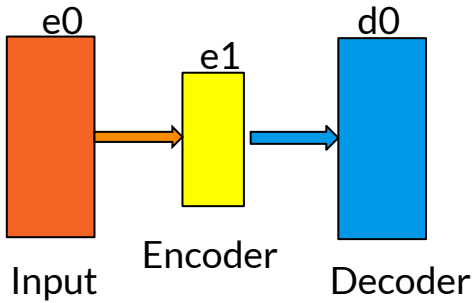


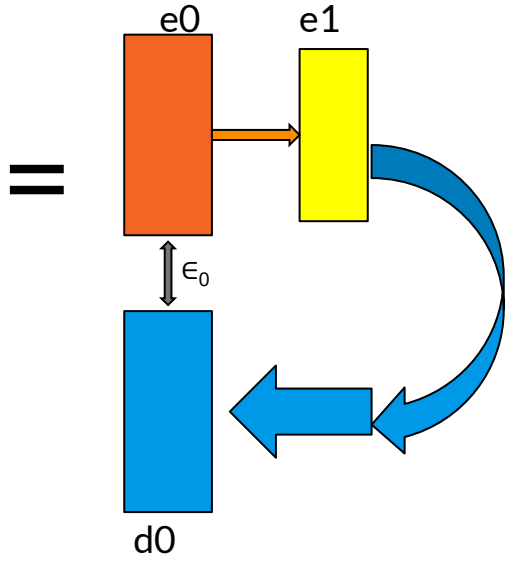
Image 2: e1,e2,e3 are the feedforward encoding layers whose activities are updated(P.C loops) using the errors(ϵ_n) produced by the corresponding d0,d1,d2 feedback generative layers that try to predict the activity of the lower e_{n-1} layers.

Image source: *Predictive coding feedback results in perceived illusory contours in a recurrent neural network*

Implementation : Predictive Coding Feedback Network

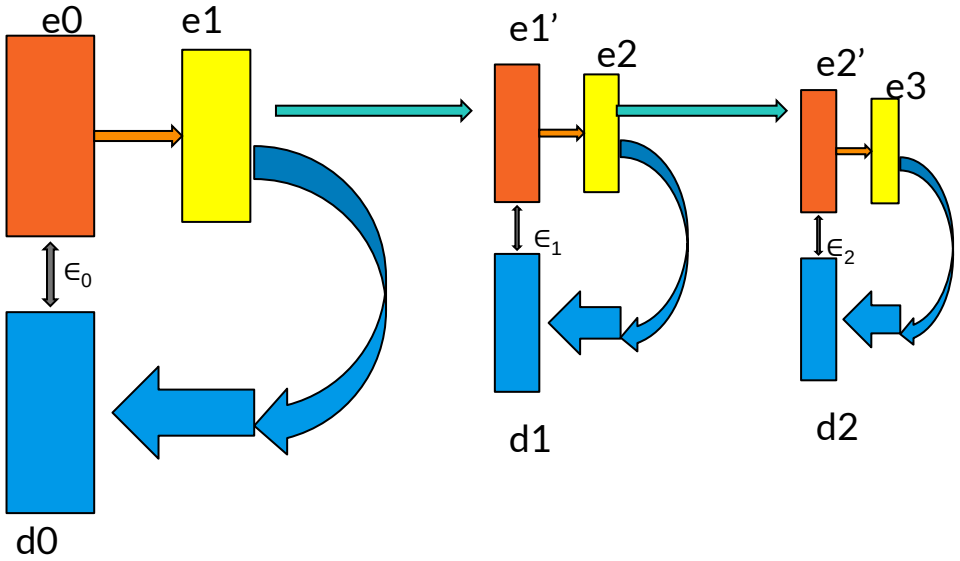


This is an autoencoder. We can rearrange this autoencoder to look something like in the next image. (Our goal is to regenerate the input at the decoder layer and reduce loss)



The autoencoder in previous slide can be hierarchically stacked

In each layer the decoding layer dn, which gets input from higher encoding layer e(n+1) tries to predict the output of lower encoding layer e(n). And the activations of higher layers are updated by the prediction error.



Intuition

This idea of prediction from a higher layer of what could be the output of lower layer is claimed to product expectations.

The higher layers when they see corners of a square expect to see a complete square from the lower layer.

If there is no complete square in the lower layer, the higher layer is suggested to change its activity. (Updation of weights based on prediction errors)

So in Kanizsa square when we see only corners that resemble a square our brain is expecting to see a full square.

Formula for activation updation

$$d_n(t+1) = \left[W^b_{n+1,n} e_{n+1}(t)\right]_+$$

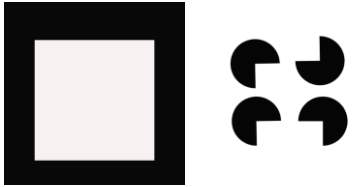
$$e_n(t+1) = \beta_n \left[W^f_{n-1,n} e_{n-1}(t+1)\right]_+ + \lambda_n d_n(t+1) + (1-\beta_n-\lambda_n) e_n(t) - \alpha_n \frac{\partial \epsilon_{n-1}(t)}{\partial e_n(t)} \quad (1)$$

Source : [Predictive coding feedback results in perceived illusory contours in a recurrent neural network]

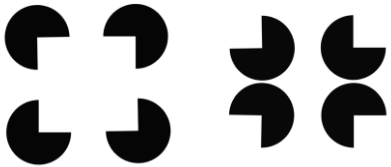
Project Details

- The project is implemented using python programming language
- Tensorflow keras library is used for encoders and classification head
- Image processing is done using OpenCv library
- The project is available on this link: https://github.com/SagarDollin/Computer_Vision_to_perceive_illusion

Training and Testing



Training set



Testing set

- First, the prediction coding feedback network is trained on Cifar100 natural images
- Then we add the classification head on top of Predictive coding and train the model on real squares and Pacmans in random orientations.
- Next we test our model on Kanizsa illusion(Pacman facing inside) to check whether the model perceives illusion
- We also test it on Out facing pacmans to verify that this model can still recognize pacmans