Redes Neurais (Deep Learning) : arquiteturas referenciais e exemplos

Março de 2021

Ref.: Cap. 6 do livro texto

A estrutura matemática das redes neurais

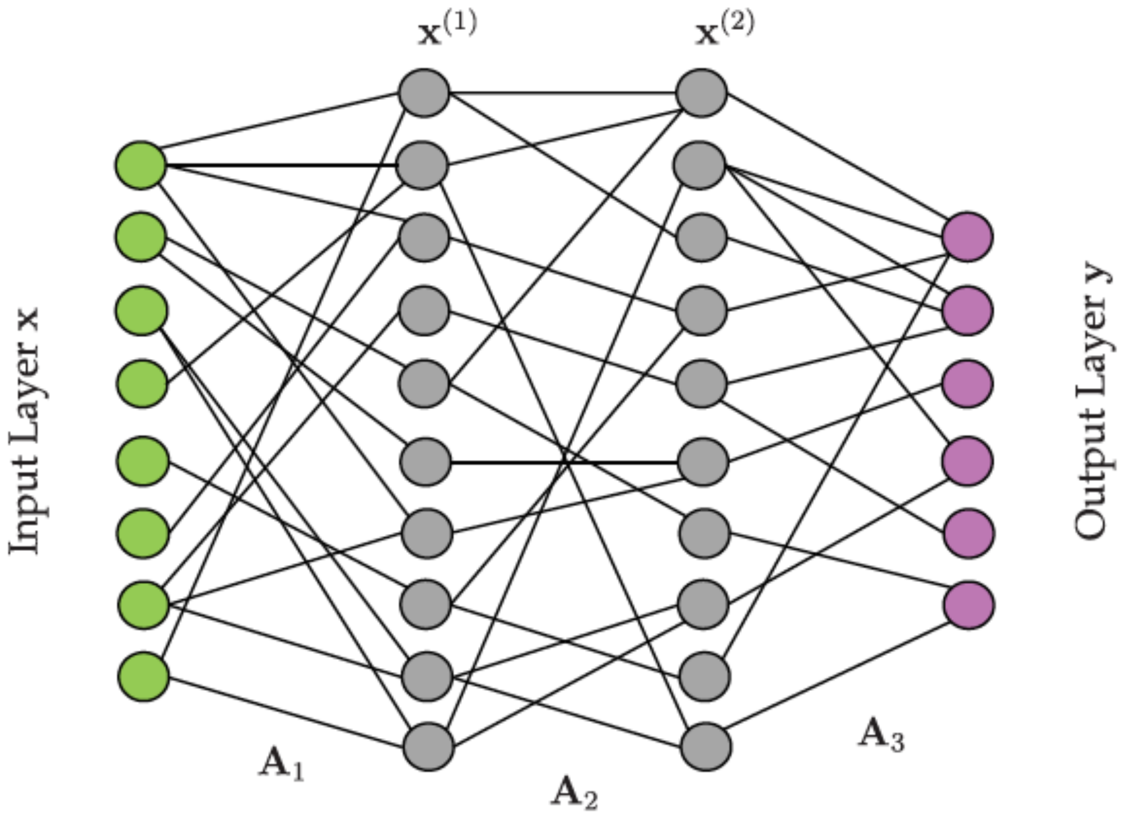


Figure 6.1 Illustration of a neural net architecture mapping an input layer $\mathbf{x}$ to an output layer $\mathbf{y}$. The middle (hidden) layers are denoted $\mathbf{x}^{(j)}$ where j determines their sequential ordering. The matrices $\mathbf{A}_j$ contain the coefficients that map each variable from one layer to the next. Although the dimensionality of the input layer $\mathbf{x} \in \mathbb{R}^n$ is known, there is great flexibility in choosing the dimension of the inner layers as well as how to structure the output layer. The number of layers and how to map between layers is also selected by the user. This flexible architecture gives great freedom in building a good classifier.

$$\mathbf{y} = f(\mathbf{x}; \mathbf{w})$$

$$f : \mathbb{R}^n \to \mathbb{R}^m$$

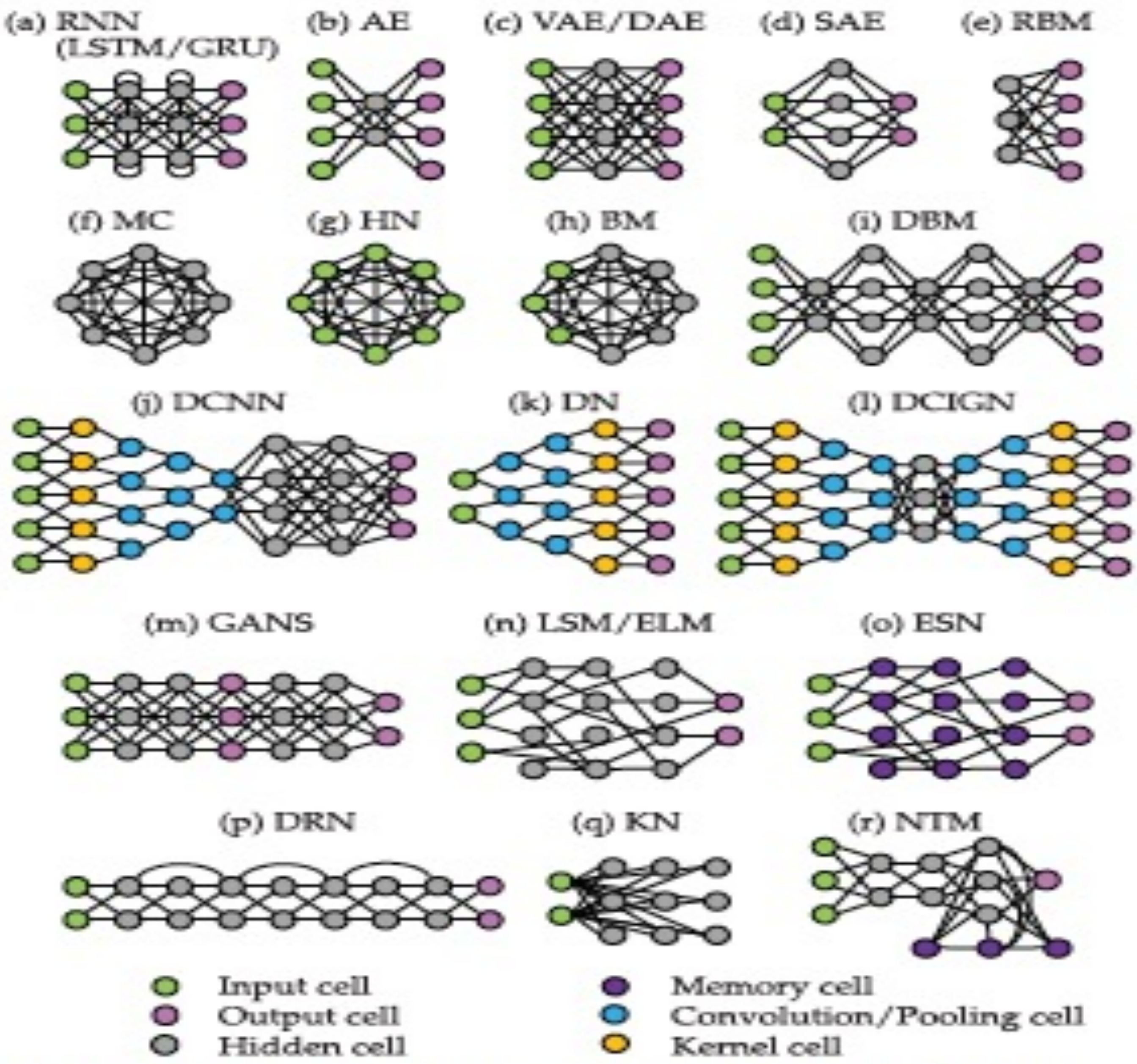


Figure 6.18 Neural network architectures commonly considered in the literature. The NNs are comprised of input nodes, output nodes, and hidden nodes. Additionally, the nodes can have memory, perform convolution and/or pooling, and perform a kernel transformation. Each network, and their acronym is explained in the text.

Arquiteturas usuais (referenciais)

- Perceptron
- Feed Forward (FF)
- Recurrent Neural Network (RNN): sistemas dinâmicos
- Auto Encoder (AE … VAE) : redução de dimensionalidade
- Markov Chain (MC)
- Hopfield network (HN)
- Boltzmann Machine (BM)
- Restricted Boltzmann Machine

Continuando..

- Deep Belief Network (DBN)
- Deep Convolutional Neural Network (DCNN)
- Deconvolutional Network (DN)
- Generative Adversarial Networks (GAN) : pra além dos dados iniciais...

Construindo arquiteturas com camadas convolucionais

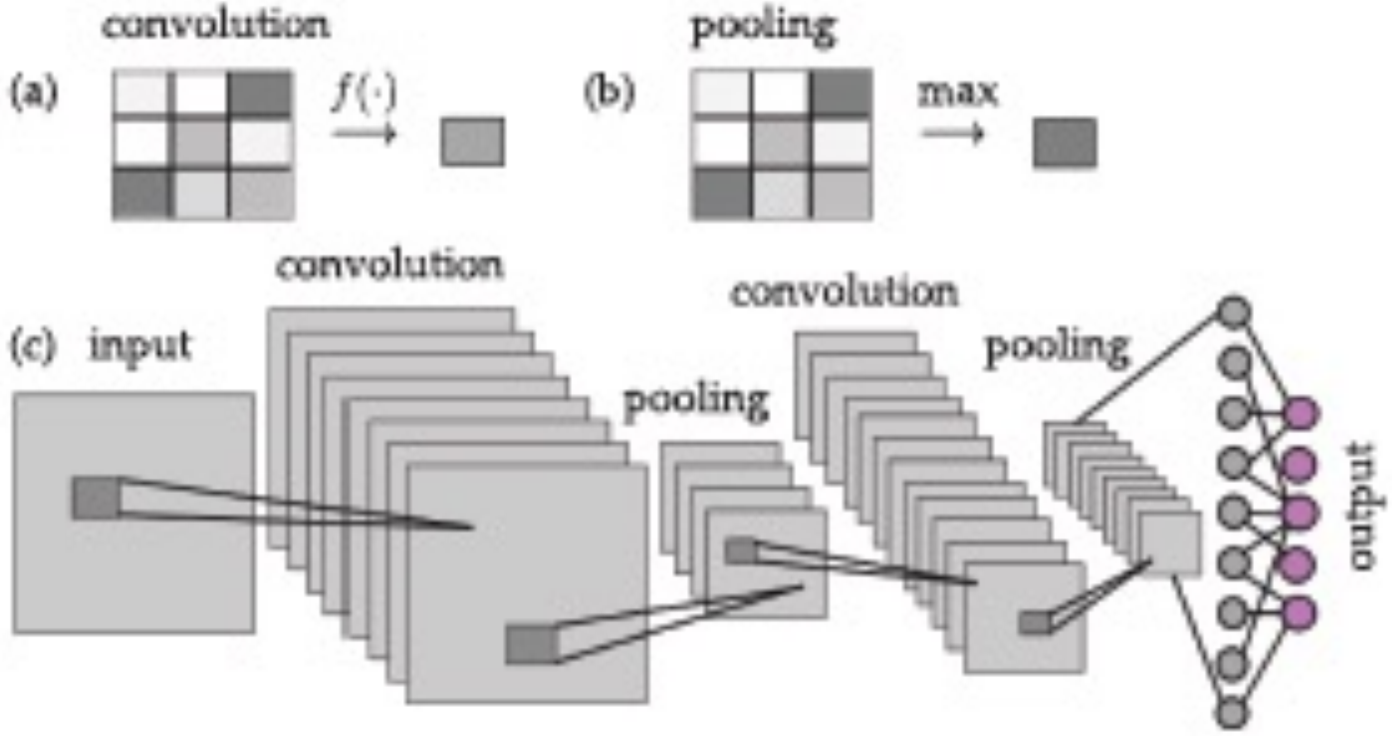
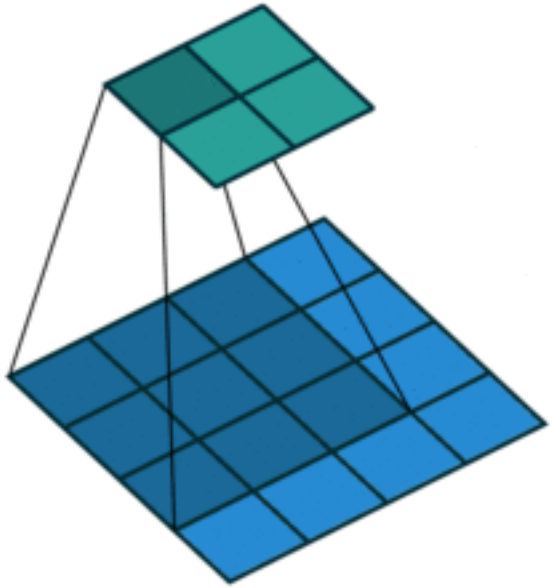
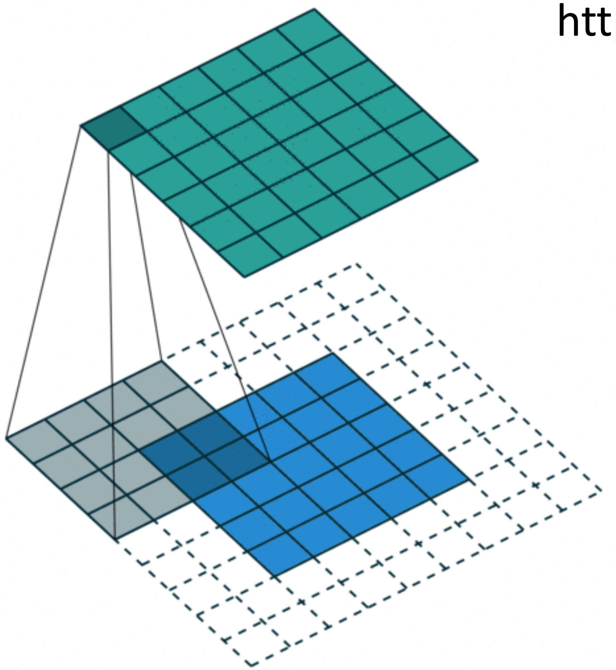
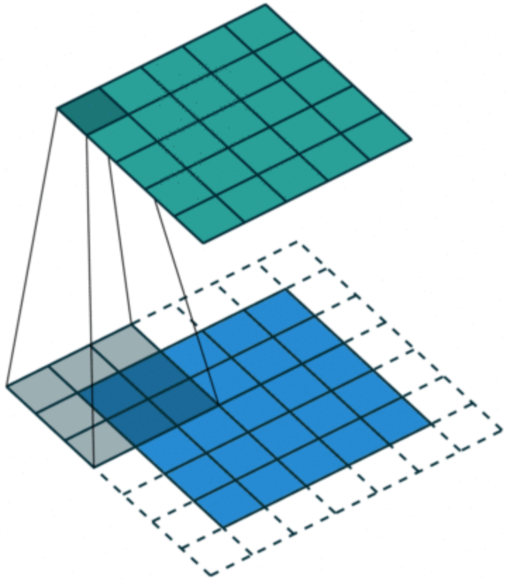


Figure 6.12 Prototypical DCNN architecture which includes commonly used convolutional and pooling layers. The dark gray boxes show the convolutional sampling from layer to layer. Note that for each layer, many functional transformations can be used to produce a variety of feature spaces. The network ultimately integrates all this information into the output layer.

https://github.com/vdumoulin

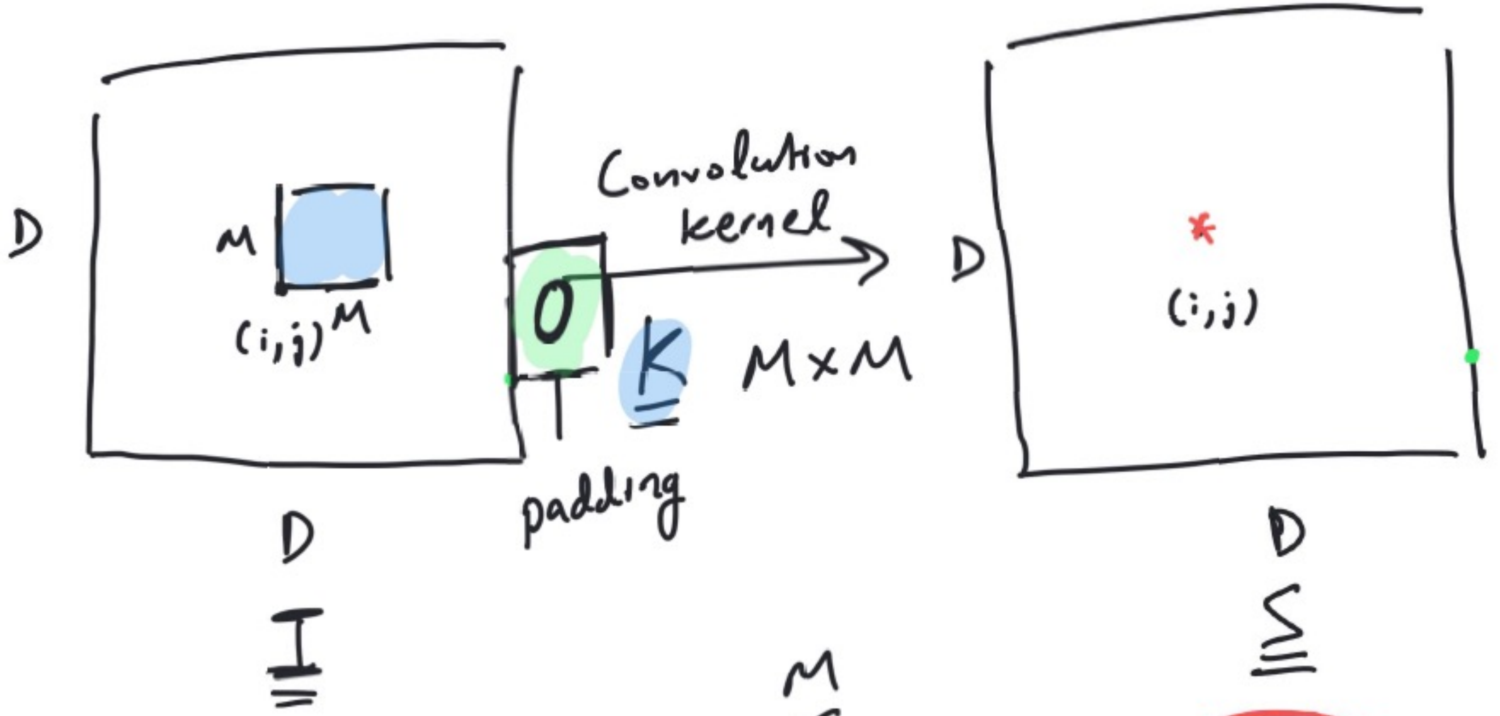






Padding and strides

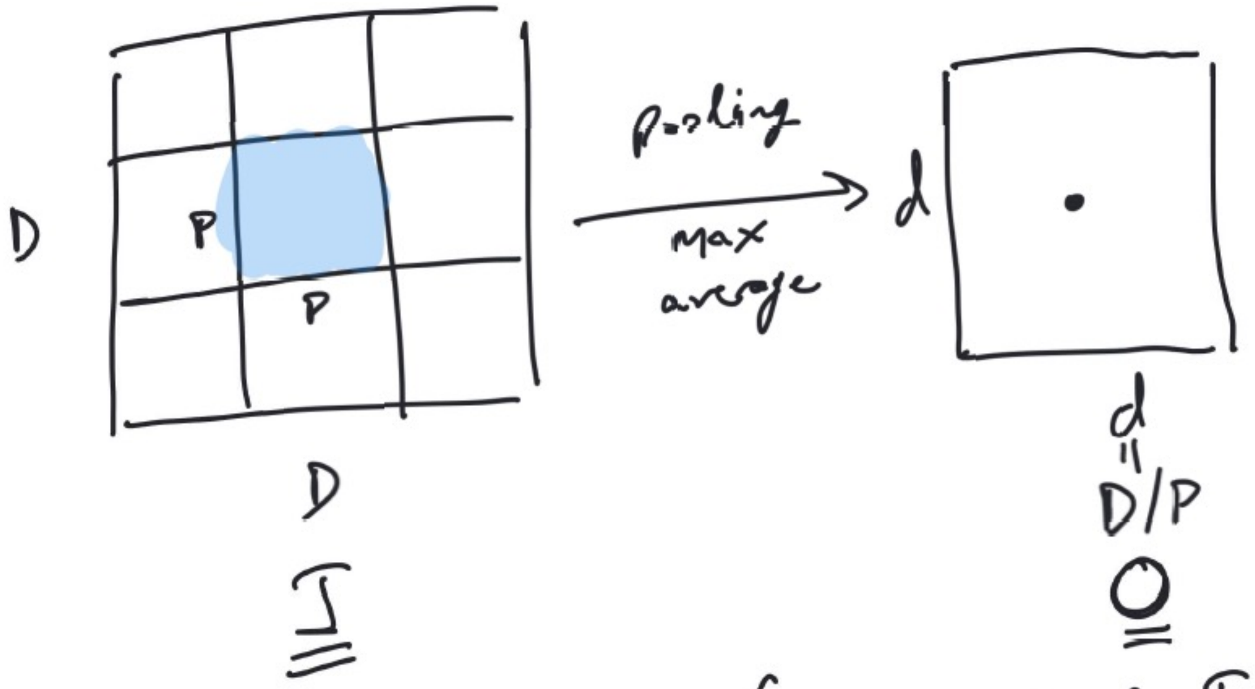
The convolutional layer



$$S_{ij} = \sum_{m,n=1}^{M} I_{i+m,j+n} K_{mn}$$

kernel parameters

The pooling layer (max)



$$\underline{\underline{O}}_{ij} = \text{pooling}\left(ij \text{ subset of } \underline{\underline{I}} \right)$$

Deep Learning . Para além de gatos e cachorros

$$\mathbf{u}(\mathbf{s}) = -K(\mathbf{s})\nabla p(\mathbf{s}), \qquad \mathbf{s} \in \mathcal{S},$$
$$\nabla \cdot \mathbf{u}(\mathbf{s}) = f(\mathbf{s}), \qquad \mathbf{s} \in \mathcal{S},$$
$$\mathbf{u}(\mathbf{s}) \cdot \hat{\mathbf{n}}(\mathbf{s}) = 0, \qquad \mathbf{s} \in \partial\mathcal{S},$$
$$\int_{\mathcal{S}} p(\mathbf{s}) d\mathbf{s} = 0,$$

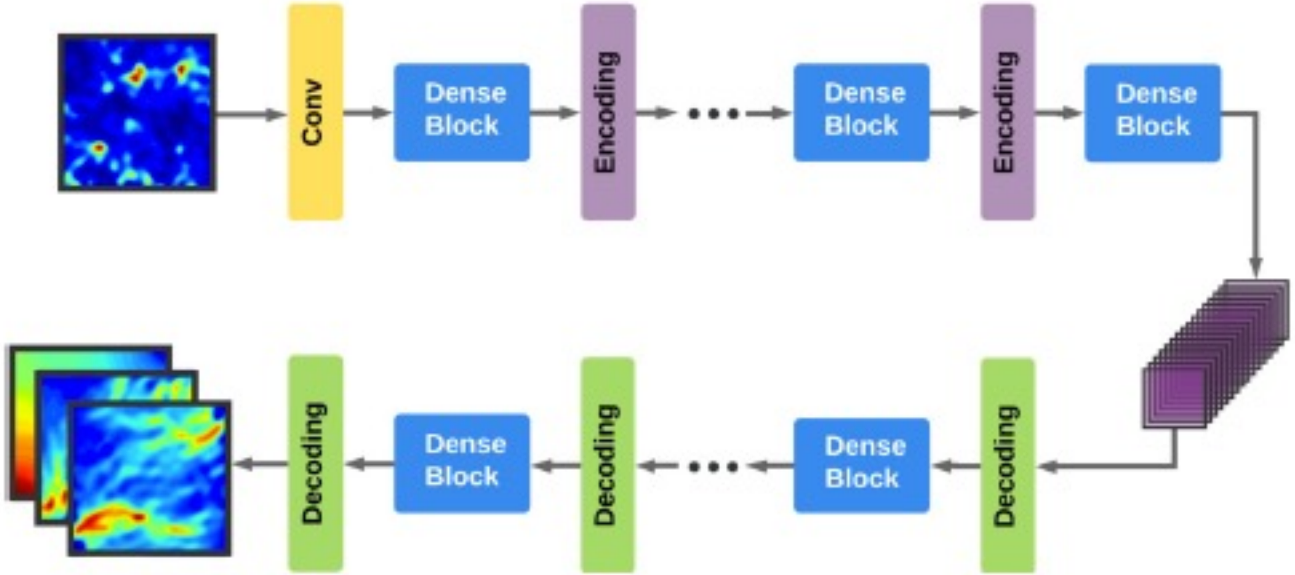


Fig. 3. Network architecture: DenseED.

Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification
Y Zhu, N Zabaras
Journal of Computational Physics 366, 415-447