3 Physical Background

3.1 S-Matrix Calculus

To implement the desired algorithm we need to be able to calculate the optical behavior of stacked Meta Surfaces. The mathematical framework we will use is called Scatter Matrix Calculus and this section will give some insight into its physical origin and how to use it. We will start at the very beginning with the Maxwell Equations in matter.

Maxwell Equations

$$\nabla \times \mathbf{E}(\mathbf{r},t) = -\frac{\partial}{\partial t}\mathbf{B}(\mathbf{r},t) \tag{3.1}$$

$$\nabla \cdot \mathbf{D}(\mathbf{r},t) = \rho_{\text{ext}}(\mathbf{r},t) \tag{3.2}$$

$$\nabla \times \mathbf{H}(\mathbf{r},t) = \mathbf{j}(\mathbf{r},t) + \frac{\partial}{\partial t}\mathbf{D}(\mathbf{r},t) \tag{3.3}$$

$$\nabla \cdot \mathbf{B}(\mathbf{r},t) = 0 \tag{3.4}$$

The four involved fields are: $\mathbf{E}$...electric field, $\mathbf{B}$...magnetic flux density, $\mathbf{D}$...electric flux density and $\mathbf{H}$...magnetic field and the sources are the external charges ρ_{ext} and the macroscopic currents j. All the material properties are captured by the $\mathbf{D}$ and $\mathbf{H}$ fields which are defined as:

$$\begin{aligned}\mathbf{D}(\mathbf{r},t) &= \varepsilon_0\mathbf{E}(\mathbf{r},t) + \mathbf{P}(\mathbf{r},t)\\ \mathbf{H}(\mathbf{r},t) &= \frac{1}{\mu_0}[\mathbf{B}(\mathbf{r},t) - \mathbf{M}(\mathbf{r},t)]\end{aligned} \tag{3.5}$$

Where $\mathbf{P}$ is the dielectric polarization and $\mathbf{M}$ is the magnetic polarisation. One can read equation (3.5) in the following way: When the electric field $\mathbf{E}$ interacts with matter it exerts a force on all its charges and displaces them by a small amount. The separation of charges results in a counter field $\mathbf{P}$ and the total field $\mathbf{D}$ is now a superposition of $\mathbf{E}$ and $\mathbf{P}$. This set of equations describes the whole electromagnetic spectrum, in this work however we are only interested in visible (VIS) and near infra red (NIR) light so we can make some simplifications. $\rho_{\text{ext}} = 0$ and $\mathbf{M} = 0$ **1.why?**. Inserting these assumptions into the maxwell equation gives:

$$\nabla \times \mathbf{E}(\mathbf{r},t) = -\mu_0\frac{\partial}{\partial t}\mathbf{H}(\mathbf{r},t) \tag{3.6}$$

$$\varepsilon_0\nabla \cdot \mathbf{E}(\mathbf{r},t) = -\nabla \cdot \mathbf{P}(\mathbf{r},t) \tag{3.7}$$

$$\nabla \times \mathbf{H}(\mathbf{r},t) = \mathbf{j}(\mathbf{r},t) + \frac{\partial}{\partial t}\mathbf{P}(\mathbf{r},t) + \varepsilon_0\frac{\partial}{\partial t}\mathbf{E}(\mathbf{r},t) \tag{3.8}$$

$$\nabla \cdot \mathbf{H}(\mathbf{r},t) = 0 \tag{3.9}$$

Light in Vacuum

Now we can derive the famous wave equation by considering $\nabla\times$ (3.6):

$$\begin{aligned}&\nabla \times \left[\nabla \times \mathbf{E}\right] = \nabla \times \left[-\mu_0\frac{\partial}{\partial t}\mathbf{H}\right]\\ \Leftrightarrow\ &\nabla(\nabla \cdot \mathbf{E}) - \Delta\mathbf{E} = -\mu_0\frac{\partial}{\partial t}\nabla \times \mathbf{H} \quad \Big| \text{ subs. (3.8) and (3.7)}\\ \Leftrightarrow\ &\frac{1}{c^2}\frac{\partial^2}{\partial t^2}\mathbf{E} - \Delta\mathbf{E} = -\mu_0\frac{\partial}{\partial t}\mathbf{j} - \mu_0\frac{\partial^2}{\partial t^2}\mathbf{P} + \frac{1}{\varepsilon_0}\nabla(\nabla \cdot \mathbf{P})\end{aligned} \tag{3.10}$$

In vacuum ($\mathbf{P} = 0$ and $\mathbf{j} = 0$) the right side of this equation vanishes and we are left with $\frac{1}{c^2}\frac{\partial^2}{\partial t^2}\mathbf{E} - \Delta\mathbf{E} = 0$ which is solved by the plane wave $\mathbf{E} = \mathbf{E}_0 e^{i(\mathbf{kr}-\omega t)}$ where $\frac{\omega}{k} = c$. This describes the propagation of light through empty space: a sinusoidal occilation in time and space along the $\mathbf{k}$ direction where $\mathbf{E}$, $\mathbf{B}$ and $\mathbf{k}$ are all perpendicular to each other. **2.this needs details 3.show figure?**

Light in homogeneous, isotropic materials

The next question we can answer is how light propagates through a homogeneous and isotropic material. For us the dielectric polarization is some linear function of the electric field so $\mathbf{P}(\mathbf{r},t) = \hat{\chi}(\omega,\mathbf{r})\mathbf{E}(\mathbf{r},t)$. An isotropic material behaves the same for all orientations of $\mathbf{E}$ that means $\hat{\chi}(\omega,\mathbf{r})$ becomes a scalar

function $\chi(\omega, \mathbf{r})$. If the material is additionally homogeneous, that is the same everywhere independent of $\mathbf{r}$, then $\boldsymbol{\nabla} \cdot \chi(\omega, \mathbf{r}) = 0$. With equation (3.7) this gives us $\boldsymbol{\nabla} \cdot \mathbf{P} = 0$ and the wave equation simplifies to **4.why is j=0**:

$$\begin{aligned} \frac{\varepsilon}{c^2} \frac{\partial^2}{\partial t^2} \mathbf{E} - \Delta \mathbf{E} = 0 \\ \text{where} \quad \varepsilon \, \mathbf{E} := (1 + \chi)\mathbf{E} = \mathbf{E} + \mathbf{P} \end{aligned} \tag{3.11}$$

That means light behaves in these materials exactly as it would in vacuum we only have to account for a decreased speed of light $c' = \frac{c}{\sqrt{\varepsilon}} =: \frac{c}{n}$ with the refractive index n. This is equivalent to a decreased wavelength $\lambda' =: \frac{\lambda}{n}$ which results in an increased wave vector $k' = \frac{2\pi}{\lambda'} = n\,k$. A complex valued $n := \eta + i\kappa$ even captures the possibility of a decaying field:

$$\mathbf{E} = \mathbf{E}_0 e^{i(n\mathbf{kr} - \omega t)} = \mathbf{E}_0 \underbrace{e^{-\kappa \mathbf{kr}}}_{\text{decay}} \underbrace{e^{i(\eta \mathbf{kr} - \omega t)}}_{\text{oscillation}} \tag{3.12}$$

Interfaces

The meta surface stacks we want to understand are obviously not one homogeneous material. Rather they contain many interfaces between different materials and we can again use the maxwell equations to predict how light will behave at such an interface.

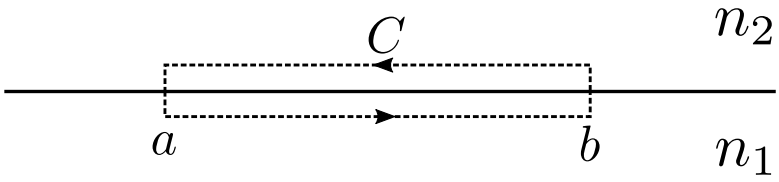


Figure 1: An interface of two materials with different refractive indices n_1 and n_2

Let us consider a closed contour C around an interface as seen in Figure 1 and integrate the first maxwell equation (3.6) over the surface A enclosed by that contour:

$$\begin{aligned} \int_A \boldsymbol{\nabla} \times \mathbf{E}(\mathbf{r}, t)\, \mathrm{d}\mathbf{A} &= -\mu_0 \frac{\partial}{\partial t} \int_A \mathbf{H}(\mathbf{r}, t) \mathrm{d}\mathbf{A} \\ \overset{\text{stokes}}{\Longleftrightarrow} \qquad \int_{C=\partial A} \mathbf{E}(\mathbf{r}, t) \mathrm{d}\mathbf{r} &= -\mu_0 \frac{\partial}{\partial t} \int_A \mathbf{H}(\mathbf{r}, t) \mathrm{d}\mathbf{A} \end{aligned} \tag{3.13}$$

But now we can bring the contour infinitely close to the interface and thus reduce the right hand side of the equation, the total magnetic flux**5.?** through the surface, to zero. That leaves us with:

$$\begin{aligned} \int_a^b \mathbf{E}_1 \, \mathrm{d}\mathbf{r} \; + \int_b^a \mathbf{E}_2 \, \mathrm{d}\mathbf{r} &= 0 \\ \Longleftrightarrow \qquad \int_a^b \mathbf{E}_1 \, \mathrm{d}\mathbf{r} &= \int_a^b \mathbf{E}_2 \, \mathrm{d}\mathbf{r} \end{aligned} \tag{3.14}$$

Because a and b were chosen arbitrarily that means that the transverse field components along the path need to be continuous so $\mathbf{E}_1^{\parallel} = \mathbf{E}_2^{\parallel}$. The analog expression $\mathbf{H}_1^{\parallel} = \mathbf{H}_2^{\parallel}$ can be shown by starting with the first maxwell equation (3.6) instead.

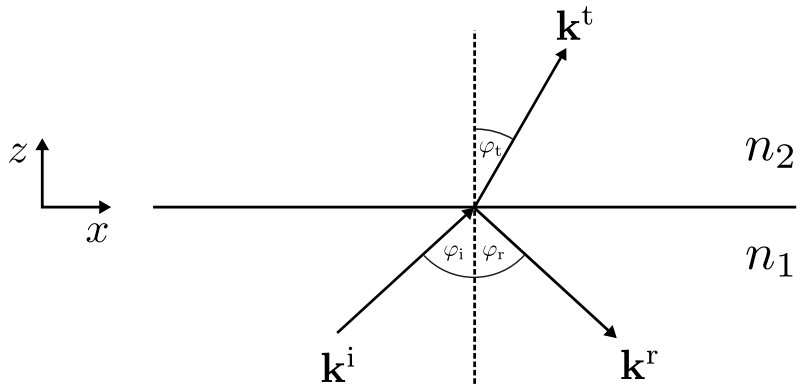


Figure 2: Interface between two materials of different refractive indices n_1 and n_2. Shown is an incident wave with wave vector $\mathbf{k}_{\mathrm{i}}$ which is partially transmitted and partially reflected.

Let us now consider the same interface from before and have an incident field $\mathbf{E}^{\mathrm{i}}$ interact with it. From figure 2 we can see:

$$\mathbf{k}^{\mathrm{i}} = k_0 n_1 \begin{pmatrix} -\sin\varphi_{\mathrm{i}} \\ 0 \\ -\cos\varphi_{\mathrm{i}} \end{pmatrix}, \quad \mathbf{k}^{\mathrm{r}} = k_0 n_1 \begin{pmatrix} \sin\varphi_{\mathrm{r}} \\ 0 \\ -\cos\varphi_{\mathrm{r}} \end{pmatrix}, \quad \mathbf{k}^{\mathrm{t}} = k_0 n_2 \begin{pmatrix} \sin\varphi_{\mathrm{t}} \\ 0 \\ \cos\varphi_{\mathrm{t}} \end{pmatrix} \tag{3.15}$$

Now we can decompose the electric field into a component were $\mathbf{E} = E\,\vec{\mathbf{e}_y}$ called transverse electric (TE) and its orthogonal component where $\mathbf{H} = H\,\vec{\mathbf{e}_y}$ called transverse magnetic (TM). If we apply the continuity condition from before to the TE component we get $\mathbf{E}_1 = \mathbf{E}_2$ at $z = 0$:

$$\begin{aligned} \mathbf{E}^{\mathrm{i}} + \mathbf{E}^{\mathrm{r}} &= \mathbf{E}^{\mathrm{t}} \\ E^{\mathrm{i}} e^{i(k_x^{\mathrm{i}} x)} + E^{\mathrm{r}} e^{i(k_x^{\mathrm{r}} x)} &= E^{\mathrm{t}} e^{i(k_x^{\mathrm{t}} x)} \end{aligned} \tag{3.16}$$

This is only possible for all x if:

$$E^{\mathrm{i}} + E^{\mathrm{r}} = E^{\mathrm{t}} \quad \text{and} \quad k_x^{\mathrm{i}} = k_x^{\mathrm{r}} = k_x^{\mathrm{t}} \tag{3.17}$$

Two basic laws of optics lie in this relation: The law of reflection **6.fix the signs**

$$k_x^{\mathrm{i}} = k_x^{\mathrm{r}} \quad \Rightarrow \quad -k_0 n_1 \sin\varphi_i = k_0 n_1 \sin\varphi_r \quad \Rightarrow \quad \varphi_i = \varphi_r := \varphi_1 \tag{3.18}$$

and the Snells law of refraction

$$k_x^{\mathrm{i}} = k_x^{\mathrm{t}} \quad \Rightarrow \quad -k_0 n_1 \sin\varphi_i = k_0 n_2 \sin\varphi_t \quad \Rightarrow \quad n_1 \sin\varphi_i = n_2 \sin\varphi_t \tag{3.19}$$

Fresnel Equations

To fully describe an interface we also need know the fraction of the transmitted $t := \frac{\mathbf{E}^t}{\mathbf{E}^i}$ and the reflected $r := \frac{\mathbf{E}^r}{\mathbf{E}^i}$ fields. Again using the TE polarization we get that $\mathbf{E} = E\,\vec{\mathbf{e}}_y$ and $\mathbf{H}$ is perpendicular to $\mathbf{E}$ in the x-z plane. So

$$\mathbf{H}^{\mathrm{i}} = H^{\mathrm{i}} \begin{pmatrix} -\cos\varphi_{\mathrm{i}} \\ 0 \\ -\sin\varphi_{\mathrm{i}} \end{pmatrix}, \quad \mathbf{H}^{\mathrm{r}} = H^{\mathrm{r}} \begin{pmatrix} \cos\varphi_{\mathrm{r}} \\ 0 \\ \sin\varphi_{\mathrm{r}} \end{pmatrix}, \quad \mathbf{H}^{\mathrm{t}} = H^{\mathrm{t}} \begin{pmatrix} -\cos\varphi_{\mathrm{t}} \\ 0 \\ \sin\varphi_{\mathrm{t}} \end{pmatrix} \tag{3.20}$$

The H_x component is tangent to the interface so it also needs to be continuous across the boundary:

$$\begin{aligned} & \mathbf{H}_x^{\mathrm{i}} + \mathbf{H}_x^{\mathrm{r}} = \mathbf{H}_x^{\mathrm{t}} \\ \iff \quad & H^{\mathrm{i}} \cos\varphi_1 - H^{\mathrm{r}} \cos\varphi_1 = H^{\mathrm{t}} \cos\varphi_2 \end{aligned} \tag{3.21}$$

At last maxwells first equation (3.6) allows us connect the magnitudes of $\mathbf{H}$ and $\mathbf{E}$ via the refractive index $H \sim n\,E$:

$$n_1 E^{\mathrm{i}} \cos\varphi_1 - n_1 E^{\mathrm{r}} \cos\varphi_1 = n_2 E^{\mathrm{t}} \cos\varphi_2 \tag{3.22}$$

Substituting E^{r} or E^{i} with (3.17) and rearranging we get:

$$t = \frac{2n_1 \cos\varphi_1}{n_1 \cos\varphi_1 + n_2 \cos\varphi_2} \quad \text{and} \quad r = \frac{n_1 \cos\varphi_1 - n_2 \cos\varphi_2}{n_1 \cos\varphi_1 + n_2 \cos\varphi_2} \tag{3.23}$$

These are called the Fresnel equations for the TE component. The TM component can be treated analogous which yields:

$$t = \frac{2n_1 \cos\varphi_1}{n_2 \cos\varphi_1 + n_1 \cos\varphi_2} \quad \text{and} \quad r = \frac{n_2 \cos\varphi_1 - n_1 \cos\varphi_2}{n_2 \cos\varphi_1 + n_1 \cos\varphi_2} \tag{3.24}$$

For perpendicular incident light at $\varphi = 90°$ these should describe the same situation as we can no longer differentiate between TE and TM components and indeed for this angle they are equivalent **7.no they are not**:

$$t = \frac{2n_1}{n_1 + n_2} \quad \text{and} \quad r = \frac{n_1 - n_2}{n_1 + n_2} \tag{3.25}$$

3.2 Neural Networks

Artificial Neural Networks (ANN's or short NN's) are a kind of data structure inspired by the biological neurons found in nature. They can be used to find a wide range of input output relations. One classic example is mapping pictures of hand written digits to the actual digits. Rather then explicitly programmed NN's are trained on a dataset (X, Y) of correct input output pairs.

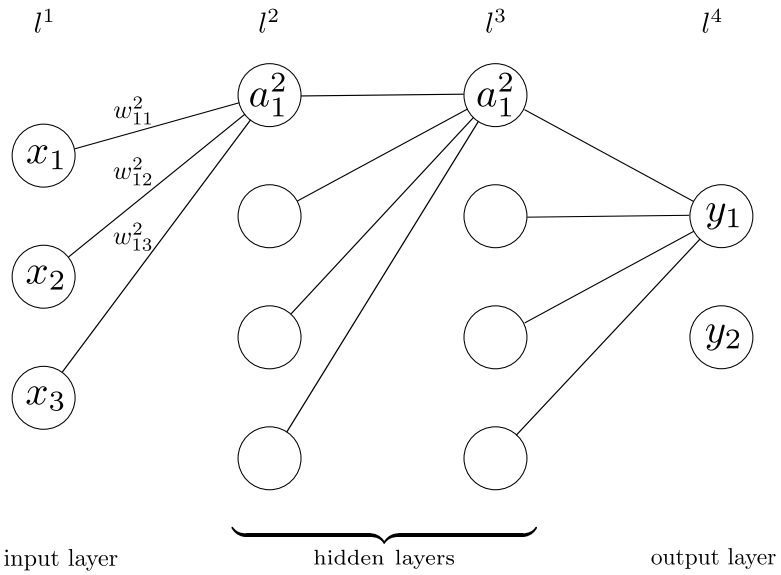


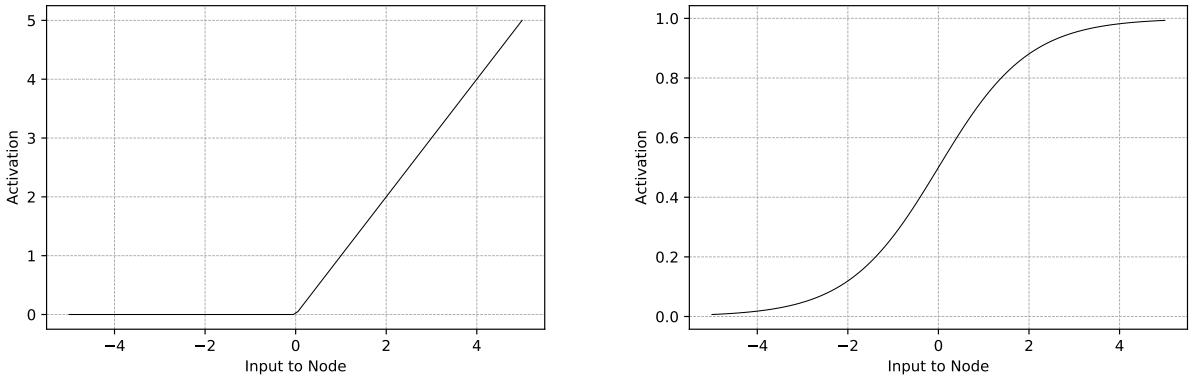
Figure 3: The most simple kind of NN is called densely connected or multilayer perceptron. For clarity only connections to the top most node of each layer are shown.

Multilayer Perceptron This kind of classic NN consist of single nodes which are organized into layers. Every node is connected to all the nodes of the previous and next layer thus they are called dense. Each node holds a value called activation a where the activation to the first layer is the input to the network, here: (x_1, x_2, x_3). To calculate the activation of a node one has to multiply all the activations of the previous layer with their respective weights w, add the bias b and finally apply a non-linear activation function σ. For the index notation superscripts specify the layer and subscripts the node. So a_1^2 is the activation of the first node in the second layer. To characterize a weight two subscripts are needed for the end and beginning of the connection. For the example in figure 3 that means:

$$a_1^2 = \sigma\left(\sum_i w_{1i}^2\, x_i + b_1^2\right) \tag{3.26}$$

However it is more convenient to stop considering every node individually and to view the involved quantities as vectors and matrices. So that (3.26) can be written as:

$$\mathbf{a}^l = \sigma\left(\hat{\mathbf{w}}^l \mathbf{a}^{l-1} + \mathbf{b}^l\right) \tag{3.27}$$



(a) ReLu (b) sigmoid

Figure 4: Two examples of activation functions σ. Especially the relu function has a destinct on and off state similar to a biological neurons.

Training During training the network output $\mathrm{NN}(\mathbf{x}) := \mathbf{y}'$ is calculated though repeated use of (3.27) and is then compared with the known correct output $\mathbf{y}$ by a cost function $C = C(\mathbf{y}, \mathbf{y}')$. This might simply be the mean squared difference between $\mathbf{y}$ and $\mathbf{y}'$:

$$C_{\mathrm{mse}}(\mathbf{y}, \mathbf{y}') = \sum_i \left(y_i - y_i'\right)^2 \tag{3.28}$$

but there are more sophisticated cost functions for different kind of outputs. Now we can quantify how well the NN is performing but how should the weights and biases be changed to improve this performance? Here the very important Algorithm *Backpropagation* is used and allows a efficient calculation of $\boldsymbol{\nabla} C_{\mathbf{b}^l}$ and $\boldsymbol{\nabla} C_{\hat{\mathbf{w}}^l}$. These are used to gradually change the weights and biases to minimize the cost function. A very comprehensive explanation of Backpropagation can be found here: [**?**].

Convolutional Neural Networks An area where NNs have been very successful is image recognition or more general computer vision but the described multilayer perceptron has a number of weaknesses for this kind of task. Let's say our input is a n by n, gray scale image. This can be expressed as a $n \times n$ matrix, flattened and fed into the input layer (see figure 5). But now the number of weights to the next layer $\hat{\mathbf{w}}^2$ is $n \cdot n \cdot l^2$ which soon becomes unfeasible.

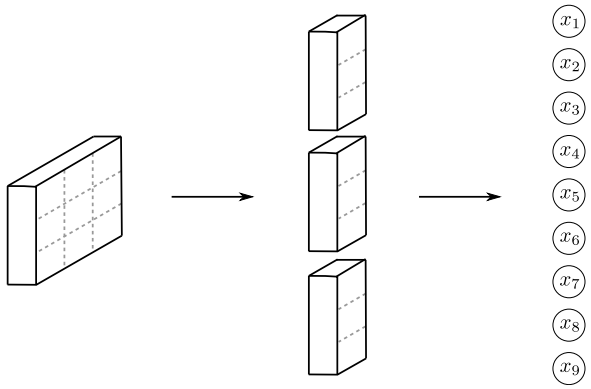


Figure 5: Flattening of a 3×3 matrix to fit the input of a multilayer perceptron.

Computational limits aside there is another problem. Imagine an image with the letter T in the top right corner. If this letter moves to a different position the networks reaction will be completely different because the weights and biases involved are completely different. So the NN cannot learn the concept "letter T" independent of its position in the picture. Also the information about the distance between pixels is lost.

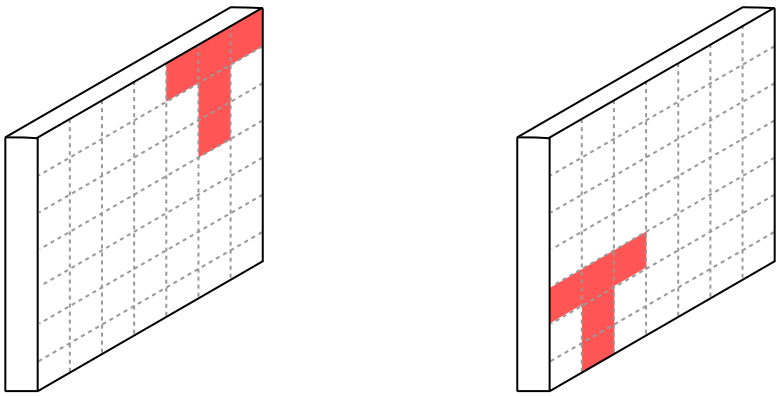


Figure 6: Two pictures of a T at different positions where the red color signifies a high value in the grayscale image. After the flatten operation seen in figure 5 very different nodes are active.

These problems led to the development of a new kind of layer called *Convolution*. A fixed size *kernel* is shifted over the matrix and at every position the the point wise product between kernel and matrix is calculated and summed. The result is called feature map of that kernel (see figure 7). Notice how the greatest value of the feature map is at the position of the letter T. So with only a small number of weights the convolution is able to detect the T independent of its position in the image. One convolutional layer contains not only a single but a number of different kernels k so that the shape of the $n \times n$ matrix transforms to $(n-1) \times (n-1) \times k$. In a Convolutional Network (ConvNet) multiple of these layers are used so that it can find "patterns in patterns". For the letter detection example one could imagine the first layer to detect various edges and the next layer to detect letters in the position of these edges.

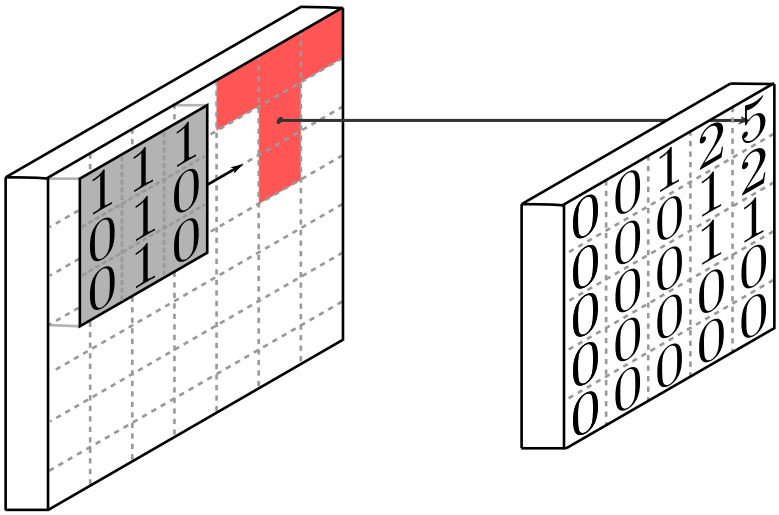


Figure 7: Example of a convolution where white pixel are 0 and red pixel are 1. The 3×3 Kernel is shifted over the image and when its directly over the The "T kernels" feature map is greatest at the position of the original letter.

Pooling Layers For a big image and a large number of kernels the output shape of a convolutional layers is still $n \times n \times k$, so quite large. Also notice how the "T kernel's" feature map is not only active at the exact position of the T but in the general region. The solution to this is to downsample the output with a *Pooling Layer*. Here a smaller kernel, usually 2×2 is shifted over the matrix two steps at a time and at every position an operation is performed to reduce the number of values to one. This could be taking the maximum or the average. This operation reduces the matrix in the x and y dimension by a factor of 2.

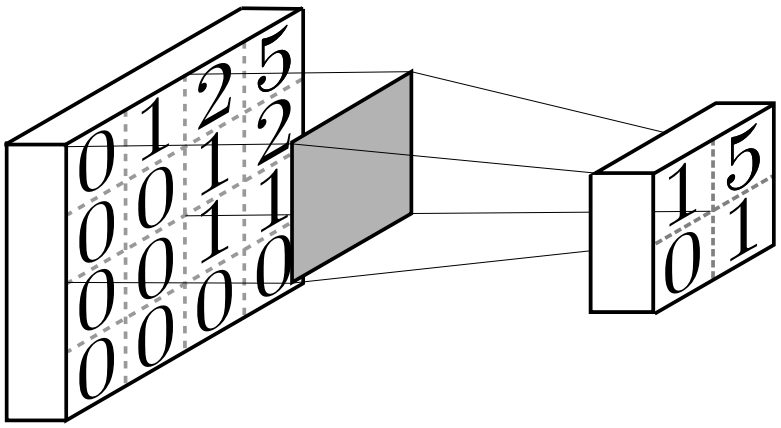


Figure 8: Example of a Max Pooling Layer. For every 2 by 2 field the maximum is calculated. After applying first the convolution and then the pooling layer the information "T in the top right corner" is still there and size of the resulting matrix is very manageable

Example Network Architecture Now all the building blocks for a complete ConvNet are available. Repeatedly alternating convolution and pooling layers changes the input from wide in x and y dimension and narrow in z to a long z-strip. At the very end this strip is fed into one densely connected layer which is in turn connected to the output neurons.

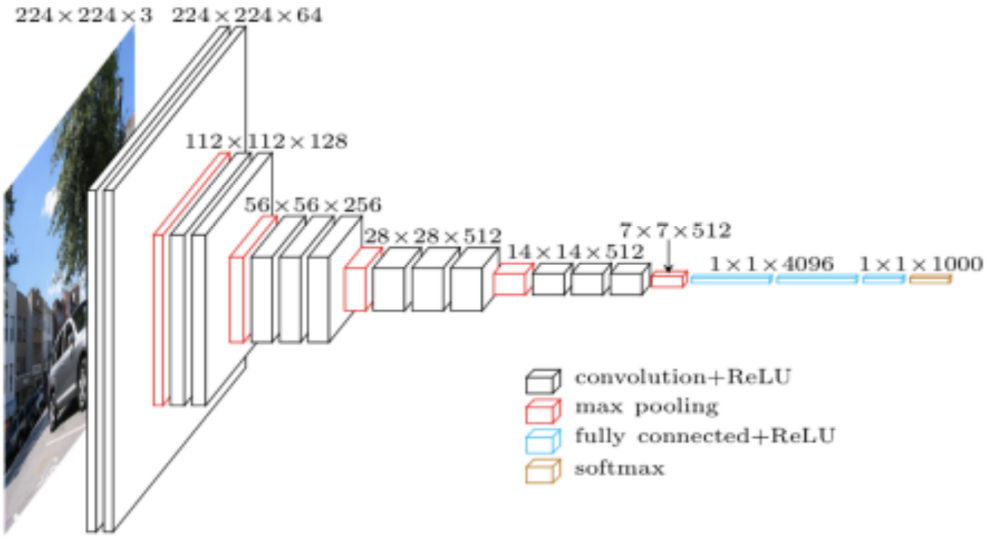


Figure 9: Example for a complete ConvNet. Note the input is in this case an RGB image so there are three layers in the z dimension corresponding to the different colors. In this case the the first layer has 64 kernels of size $3 \times 3 \times 3$. The next convolution then has 128 kernels of size $3 \times 3 \times 64$. [citation needed]

1D ConvNets The algorithm is applied to spectra, so a functions $I(\lambda)$. This data is only one dimensional but all the same ideas apply. Kernels are here sized $1 \times 3 \times z$ and are shifted in one dimension. Same goes for the pooling. These 1D convolution might detect features like rising and falling edges.

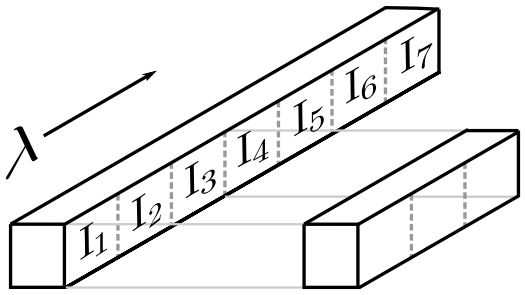
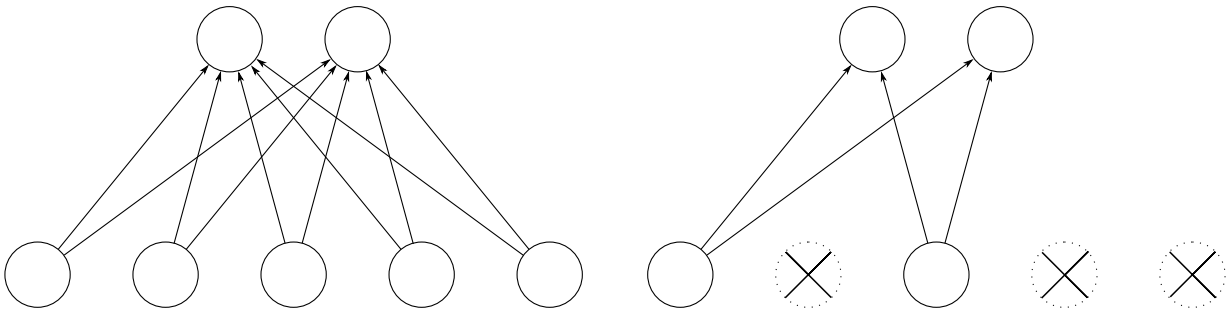


Figure 10: Example of a 1D convolution. A 1×3 kernel is shifted over a spectrum I discretized at 9 wavelengths

Dropout Layer In 2014 Srivastava et al.[?] presented a method to prevent overfitting and speed up the training process of large Neural Networks. During training they randomly drop a number of neurons in a layer along with all connections to and from these neurons. This prevents the neurons from co-adapting 8.? and because there are less weights and biases to tune for each step the training becomes overall faster.



(a) no dropout (b) with dropout

Figure 11: Example of a dropout applied to the bottom layer. Only two of the neurons remain active and only their weights and biases are modified during this training step. Figure found in [?] (modified)