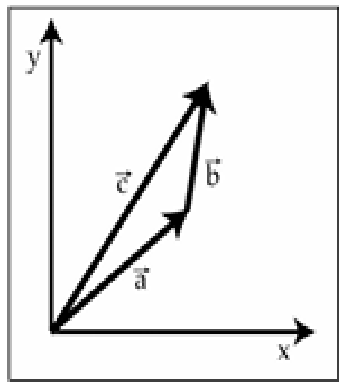
Vector addition and subtraction

Vectors can be added and subtracted. Graphically, we can think of adding two vectors together as placing two line segments end-to-end, maintaining distance and direction. An example of this is shown in the illustration, showing the addition of two vectors $\vec{a}$ and $\vec{b}$ to create a third vector $\vec{c}$.

$$\vec{a} + \vec{b} = \vec{c}$$

Numerically, we add vectors component-by-component. That is to say, we add the x components together, and then separately we add the y components together. For example, if $\vec{a} = [4,3]$ and $\vec{b} = [1,2]$, then:



$$\vec{c} = \vec{a} + \vec{b}$$

$$\vec{c} = [4,3] + [1,2]$$

$$\vec{c} = [4 + 1, 3 + 2]$$

$$\vec{c} = [5,5]$$

Similarly, in vector subtraction:

$$\vec{c} = \vec{a} - \vec{b}$$

$$\vec{c} = [4,3] - [1,2]$$

$$\vec{c} = [3,1]$$

Vector addition has a very simple interpretation in the case of things like displacement. If in the morning a ship sailed 4 miles east and 3 miles north, and then in the afternoon it sailed a further 1 mile east and 2 miles north, what was the total displacement for the whole day? 5 miles east and 5 miles north – vector addition at work.

Linear independence

If two vectors point in different directions, even if they are not very different directions, then the two vectors are said to be *linearly independent*. If vectors $\vec{a}$ and $\vec{b}$ point in the same direction, then you can multiply vector $\vec{a}$ by a constant, scalar value and get vector $\vec{b}$, and vice versa to get from $\vec{b}$ to $\vec{a}$. If the two vectors point in different directions, then this is not possible to make one out of the other because multiplying a vector by a scalar will never change the direction of the vector, it will only change the magnitude. This concept generalizes to families of more than two vectors. Three vectors are said to be linearly independent if there is no way to construct one vector by combining scaled versions of the other two. The same definition applies to families of four or more vectors by applying the same rules.

The vectors in the previous figure provide a graphical example of linear independence. Vectors $\vec{a}$ and $\vec{c}$ point in slightly different directions. There is no way to change the length of vector $\vec{a}$ and generate vector $\vec{c}$, nor vice-versa to get from $\vec{c}$ to $\vec{a}$. If, on the other hand, we consider the family of vectors that contains $\vec{a}$, $\vec{b}$ and $\vec{c}$, it is now possible, as shown, to add vectors $\vec{a}$ and $\vec{b}$ to generate vector $\vec{c}$. So the family of vectors $\vec{a}$, $\vec{b}$ and $\vec{c}$ is *not* linearly independent, but is instead said to be linearly dependent. Incidentally, you could change the length of any or all of these three vectors and they would still be linearly dependent.

Definition: A family of vectors is linearly independent if no one of the vectors can be created by any linear combination of the other vectors in the family. For example, $\vec{c}$ is linearly independent of $\vec{a}$ and $\vec{b}$ if and only if it is *impossible* to find scalar values of α and β such that $\vec{c} = \alpha\vec{a} + \beta\vec{b}$

Vector multiplication: dot products

Next we move into the world of vector multiplication. There are two principal ways of multiplying vectors, called *dot products* (a.k.a. *scalar products*) and *cross products*. The dot product:

$$d = \vec{a} \cdot \vec{b}$$

generates a scalar value from the product of two vectors and will be discussed in greater detail below. Do not confuse the dot product with the cross product:

$$\vec{c} = \vec{a} \times \vec{b}$$

which is an entirely different beast. The cross product generates a vector from the product of two vectors. Cross products so up in physics sometimes, such as when describing the interaction between electrical and magnetic fields (ask your local fMRI expert), but we'll set those aside for now and just focus on dot products in this course. The dot product is calculated by multiplying the x components, then separately multiplying the y components (and so on for z, etc… for products in more than 2 dimensions) and then adding these products together. To do an example using the vectors above:

$$\vec{a} \cdot \vec{b} = [4,3] \cdot [1,2]$$

$$\vec{a} \cdot \vec{b} = (4 * 1) + (3 * 2)$$

$$\vec{a} \cdot \vec{b} = 11$$

Another way of calculating the dot product of two vectors is to use a geometric means. The dot product can be expressed geometrically as:

$$\vec{a} \cdot \vec{b} = \|\vec{a}\| \|\vec{b}\| \cos\theta$$

where θ represents the angle between the two vectors. Believe it or not, calculation of the dot product by either procedure will yield exactly the same result. Recall, again from high school geometry, that $\cos 0° = 1$, and that $\cos 90° = 0$. If the angle between $\vec{a}$ and $\vec{b}$ is nearly $0°$ (i.e. if the vectors point in nearly the same direction), then the dot product of the two vectors will be nearly $\|\vec{a}\| \|\vec{b}\|$.

Definition: A dot product (or scalar product) is the numerical product of the lengths of two vectors, multiplied by the cosine of the angle between them.

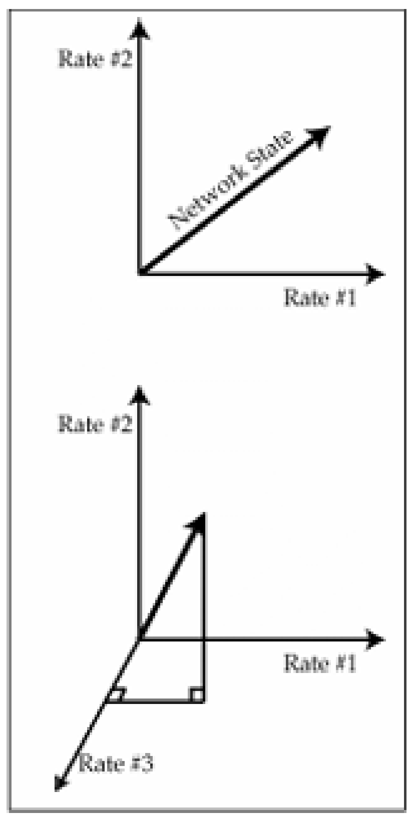
Orthogonality

As the angle between the two vectors opens up to approach $90°$, the dot product of the two vectors will approach 0, regardless of the vector magnitudes $\|\vec{a}\|$ and $\|\vec{b}\|$. In the special case that the angle between the two vectors is exactly $90°$, the dot product of the two vectors will be 0 regardless of the magnitude of the vectors. In this case, the two vectors are said to be *orthogonal*.

Definition: Two vectors are orthogonal to one another if the dot product of those two vectors is equal to zero.

Orthogonality is an important and general concept, and is a more mathematically precise way of saying "perpendicular." In two- or three-dimensional space, orthogonality is identical to perpendicularity and the two ideas can be thought of interchangeably. Whereas perpendicularity is restricted to spatial representations of things, orthogonality is a more general term. In the context of neural networks, neuroscientists will often talk in terms of two patterns of neuronal firing being orthogonal to one another. Orthogonality can also apply to functions as well as to things like vectors and firing rates. As we will discuss later in this class in the context of *fourier transforms*, the sin and cos functions can be said to be orthogonal functions. In any of these contexts, orthogonality will always mean something akin to "totally independent" and is specifically refering to two things having a dot product of zero.

Vector spaces



All vectors live within a *vector space*. A vector space is exactly what it sounds like – the space in which vectors live. When talking about spatial vectors, for instance the direction and speed with which a person is walking through a room, the vector space is intuitively spatial since all available directions of motion can be plotted directly onto a spatial map of the room.

A less spatially intuitive example of a vector space might be all available states of a neural network. Imagine a very simple network, consisting of only five neurons which we will call n_1, n_2, n_3, n_4, and n_5. At each point in time, each neuron might not fire any action potentials at all, in which case we write $n_i = 0$, where i denotes the neuron number. Alternatively, the neuron might be firing action potentials at a rate of up to 100 Hz, in which case we write that $n_i = x$, where $0 \le x \le 100$. The state of this network at any moment in time can be depicted by a vector that describes the firing rates of all five neurons:

The set of all possible firing rates for all five neurons represents a vector space that is every bit as real as the vector space represented by a person walking through a room. The vector space represented by the neurons, however, is a 5-dimensional vector space. The math is identical to the two dimensional situation, but in this case we must trust the math because our graphical intuition fails us.

If we call state 1 the state in which neuron #1 is firing at a rate of 1 Hz and all others are silent, we can write this as:

$$s_1 = [1, 0, 0, 0, 0]$$

We may further define states 2, 3, 4, and 5 as follows:

$$s_2 = [0, 1, 0, 0, 0]$$

$$s_3 = [0, 0, 1, 0, 0]$$

$$s_4 = [0, 0, 0, 1, 0]$$

$$s_5 = [0, 0, 0, 0, 1]$$

By taking combinations of these five *basis vectors*, and multiplying them by scalar constants, we can describe any state of the network in the entire vector space. For example, to generate the network state [0, 3, 0, 9, 0] we could write:

$$(3 * s_2) + (9 * s_4) = [0, 3, 0, 9, 0]$$

If any one of the basis vectors is removed from the set, however, there will be some states of the network we will be unable to describe. For example, no combination of the vectors s_1, s_2, s_3, and s_4 can describe the network state [1, 0, 5, 3, 2] without also making use of s_5. Every vector space has a set of basis vectors. The definition of a set of basis vectors is twofold: (1) linear combinations (meaning addition, subtraction and multiplication by scalars) of the basis vectors can describe any vector in the vector space, and (2) every one of the basis vectors must be required in order to be able to describe all of the vectors in the vector space. It is also worth noting that the vectors s_1, s_2, s_3, s_4, and s_5 are all orthogonal to one another. You can test this for yourself by calculating the dot product of any two of these five basis vectors and verifying that it is zero. Basis vectors are not always orthogonal to one another, but they must always be linearly independent. The vector space that is defined by the set of all vectors you can possibly generate with different combinations of the basis vectors is called the *span* of the basis vectors.