Chapter 11

Octave Translation Invariance

Octave translation invariance is a symmetry that applies both to musical scales and to individual notes within chords.

This invariance does not appear to satisfy any functional requirement. Rather, it appears to facilitate the efficient subtraction of one pitch value from another to calculate the size of the interval between them. In particular, the brain separates each pitch value into a precise pitch value modulo octaves and an imprecise absolute value, and performs subtraction separately on each of these components.

11.1 Octave Translation Invariant Aspects of Music

The following aspects of music are octave translation invariant:

- Chords and notes within chords can often be raised or lowered by an octave without significantly affecting the musical quality of a piece of music. The same applies for bass notes.
- All Western musical scales repeat themselves each octave. This rule also applies to most non-Western musical scales.
- Home chords and home notes are octave translation invariant.

- Two musical notes separated by an octave, or a whole number of octaves, have a similar perceived quality.

In all these cases we can suppose that the pitch value of musical notes is represented by the pitch value *modulo octaves*, in the sense that information about the position of the note within its octave is retained, but information about *which* octave the note is in is thrown away.

Information about octaves is not thrown away in *all* places where pitch information is processed: we know that the notes of a melody cannot be individually raised or lowered by an octave. This relates to the contour of the melody, which describes how the pitch goes up and down at different times. And, subjectively, we know that, although two notes separated by an octave sound partly the same, we can still tell that one of the notes is higher than the other.

11.2 Separation of Concerns

A common mode of operation in the brain is the separation of information into components. As previously mentioned, visual processing involves separation of information into components of position, motion, depth and colour, so that each component can be effectively processed by specialised processing areas.

We might suppose that something similar is going on with pitch: a separation into a component modulo octaves and a component that retains octave information. However, compared to other decompositions of information that occur in the brain, this particular decomposition has a rather unusual mathematical nature: an apparently simple continuum of possible pitch values is decomposed into a modulo value and a non-modulo value. What, if anything, is the point of such a decomposition?

11.3 Digital versus Analogue

How does an electronic computer represent values that can be represented as numbers from a continuum? Typically such values are represented as **floating point** values. A floating point value consists of a **mantissa**, which is a finite number of digits, and an **exponent**. In a computer the digits are normally base 2, i.e. either 0 or 1, but it will not matter too much if we pretend that they are actually decimal digits. The exponent can be thought of as telling us where the decimal point is in relation to the digits.

Examples:

- "1.023e6" means $1.023 \times 10^6 = 1{,}023{,}000$. "1.023" is the mantissa and "6" is the exponent.
- "2.54e-3" means $2.54 \times 10^{-3} = 0.00254$. "2.54" is the mantissa and "-3" is the exponent.

This floating point representation represents numbers with a certain precision determined by the number of digits. The range of values for the exponent allows for very small and very large numbers to be represented: the programmer of the computer can usually choose a standard floating point format which can represent all the numbers required to be represented and processed in their program, to a sufficient degree of accuracy for the purposes of the program.

The brain as computer must process perceptual values that, in software running on an electronic computer, would normally be represented by numbers, but, as we have already noted, the constraints of natural evolutionary design are not quite the same as those of human industrial design. In particular, the representation of numerical values in cortical maps is much more analogue than occurs in digital computers.

Firstly, there is never any recognisable division between mantissa and exponent. If the range of values required to be represented does not include very large or very small values, then there is no need for an exponent. In the cases where there is a large dynamic range (as with the perception of loudness), then the representation is effectively exponent only. This becomes a representation on a logarithmic scale.

Secondly, numerical values are not represented as finite sequences of digits. Most values are represented in terms of neurons that lie sequentially within a map, such that each neuron represents some particular value. "In-between" values are represented by means of population encoding.

Digital representations are very compact. High levels of precision can be represented in a small number of components. For example, the level of precision in human perception never exceeds 10000 values in a 1-dimensional range of values, and 4 decimal digits would be enough to store a value from a set of 10000 possible values.

In the case of pitch perception, there are about 10 octaves in the range of human hearing. Accuracy of pitch discrimination in those portions of the range with the most sensitivity (about 1000Hz to 4000Hz) is about 0.3%, or 1/240 of an octave. If this level of discrimination applied over the full range of hearing, we would be able to discriminate 2400 different pitch values. But the level of discrimination is reduced somewhat for higher and lower pitch levels, and the maximum number of distinguishable pitch values is closer to 1400.

If, at some point in the brain, the set of possible pitch values was represented by 1 neuron per pitch value, then we would need 1400 neurons to represent them. Now 1400 is not a large number of neurons. But the difficulty begins when we consider the need to calculate *relative pitch*. As we have already noted, many aspects of the perception of music are pitch translation invariant. To achieve pitch translation invariance, it is necessary, by one means or another, to compare different pitch values, and in particular to calculate the interval between two different pitch values.

A digital computer requires just 11 binary digits to represent a number from 0 to 1400. The computer can subtract an 11 bit number from an 11 bit number to get another 11 bit number (we'll ignore overflow here), using a subtraction circuit containing some small multiple of 11 bits, probably 22 or 33.

How much circuitry will it take our brain's analogue neural network to do subtraction between these values? The naïve answer is: $1400 \times 1400 = 1,960,000 \approx 2,000,000$. (I have rounded this to a simple 2,000,000, because all the numbers here are very rough.) Why so many? We need this many neurons because we need to wire up each pair of neurons representing a pair of input values to an intermediate neuron representing that particular subtraction problem, and then we need to connect each of these intermediate neurons to the corresponding neuron representing the answer. In effect the 2,000,000 neurons constitute a giant subtraction table. (Figure 11.1 shows a 4×4 subtraction table that implements subtraction of pitch values from a range of just 4 possible values, with $4 \times 4 = 16$ intermediate neurons and 7 output neurons.)

Now 2,000,000 is a non-trivial number of neurons. Perhaps not a large number in terms of the brain's total, but still large in terms of the calculation being performed. (There may also be a need for more than just one such subtraction table. We have already determined the existence of two musical cortical maps that process consonant relations between pitch values—the harmonic cortical map and the home chord cortical map—and each such map would require its own subtraction table.)

Even if providing 2,000,000 neurons is not a problem, correctly developing all the connections between the inputs and outputs and calibrating them might consume excessive resources. (More on the subject of **calibration** in the next chapter.)

In computer science terminology, we have $O(N^2)$ complexity[1] for a problem that really only requires $O(logN)$ amount of circuitry.

11.4 Digital Representations in the Brain

So, assuming that the required size of one or more subtraction tables for pitch values might impose a significant cost on the individual, can some of this complexity be reduced by using the digital solution?

To explore this possibility, I am going to analyse the problem of how to represent a series of values from 0 to 99 by separating each value into a first decimal digit and a second decimal digit.

We can assume that the original value would be represented by 100 neurons. The separate digit values would be represented by 10 neurons for the

[1] Reminder: **complexity** refers to usage of resources, *not* how complicated the problem is.

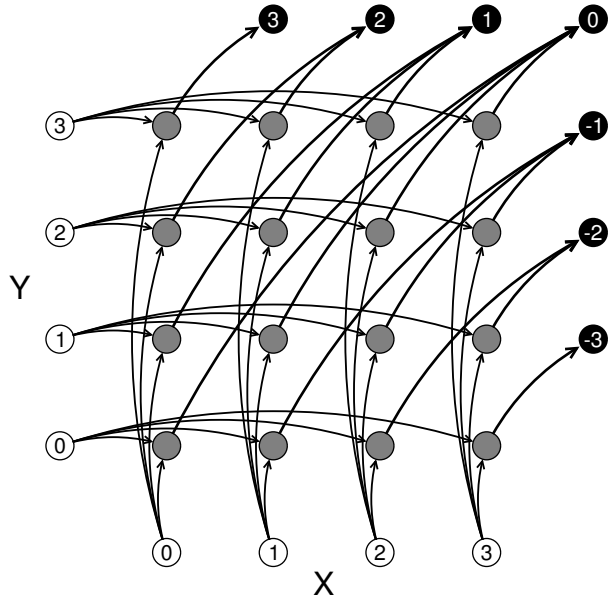

Figure 11.1. A neural subtraction table for the problem $Y - X$. Each circle represents one neuron. The white neurons are inputs representing values for X and Y. The black neurons represent the answer, and each gray neuron represents one subtraction problem. Population encoding allows the neural network to solve problems involving "in-between" values.

first digit and 10 neurons for the second digit. We have reduced the required circuitry from 100 neurons to just 20 neurons.

There is one basic problem with this simple separation, which is the general imprecision of representation of values by individual neurons. As discussed when I explained **population encoding**, each neuron represents a range of values, and each value is correspondingly represented by the activation of a range of neurons. This causes problems when we try to split the value from 0 to 99 into two values each from the range 0 to 9.

Consider, for example, a value 39.5. In the 100 neuron representation, the most active neurons will be those that maximally respond to 39 and 40, with lesser activation of those neurons active for 38 and 41, and even less for 37 and 42, and so on. No problem here: we can easily reconstruct the value 39.5 from this pattern of activity.

But now consider the separation into two digits. In the first digit, there will be neurons representing 3 and 4. Since 39.5 is in between the ranges of numbers with 3 as a first digit and 4 as a first digit, we would expect these 2 neurons to be equally active. Still no problem.

Now consider the second digit. The most active neurons will be those for 0 and 9. This represents a value between $X9$ and $Y0$, where Y is the next digit after X.

The problems begin when we try to reconstruct the original full value. The first digit is maybe 3 or 4, the second digit is maybe 9 or 0. This implies that the reconstructed number might be 39 or 40 or 30 or 49. Now 39 and 40 are good estimates, but the values of 30 and 49 are completely spurious, and nowhere near the real value.

One diagnosis of the cause of this problem is that the split of information between the first digit and the second digit is an exact split, with no sharing or overlap. This is fine in a digital computer, where the design relies on discrete components that represent discrete values with 100% reliability, but it doesn't work in neural networks where information is represented in a fuzzy manner shared between different components. If fuzzy information is to be split so that the original fuzzy information can be reliable reconstructed, then the splitting itself has to be fuzzy. This means that there has to be an overlap between what the first digit represents and what the second digit represents.

One way to do this for the 100 value example is to have the second digit be a number from 0 to 9 representing the original value modulo 10, as before, but have the first digit be a number from 0 to 19, representing the number of 5's. So 39 is represented by "79", and 40 is represented by "80".

What happens when we split and reconstruct? The reconstructed number becomes one of "70", "79", "80" or "89". In this case we still have two valid values, i.e. "79" and "80", and two spurious values "70" and "89". But this time the spurious values are intrinsically invalid, and the system can be wired to ignore them. For example, a first digit of 7 implies a number in the range from 35 to 39, and none of these numbers ends in 0, so "70" is an invalid number. Similarly a first digit of 8 implies a number in the range 40 to 45, so "89" does not represent a valid number.

This overlap between what the first digit represents and what the second digit represents introduces some redundancy, so there is less reduction in the number of neurons required. We have $20 + 10 = 30$ neurons, instead of $10 + 10 = 20$ neurons, but this is still less than 100 neurons.

We can now calculate the reduction of the size of the subtraction tables using the fuzzy split representation: ignoring the details of wrap-arounds and overflows, the original representation requires $100 \times 100 = 10000$ neurons to do subtraction, whereas the fuzzy split representation requires $20 \times 20 + 10 \times 10 = 400 + 100 = 500$ neurons, which is considerably fewer.

11.5 Split Representation of Pitch

The previous analysis suggests that the representation of pitch information is such that pitch values are split into two components:

- A pitch value modulo octaves, which has maximum precision.
- An absolute pitch value which is less precise.

Exactly how imprecise is the imprecise absolute pitch value representation? There is no obvious way to measure this, because the combined effect of the two representations is always equivalent to a representation of a single precise pitch value. From our analysis we would expect that the average error of the absolute pitch value representation is somewhat larger than the average error in the representation of the pitch value modulo octaves (because the absolute value representation is the imprecise first "digit") and somewhat smaller than an octave (because the split into two "digits" is fuzzy).

It is possible that neurological patients exist who have suffered some type of localised brain damage, and who can be identified as having lost the modulo-octaves representation of pitch. If these patients still have some degree of pitch perception, then the accuracy of their pitch discrimination could be an indicator of the accuracy of the absolute component of the split pitch value.

It might be supposed that our ability to detect up and down motions in pitch is tied to the absolute imprecise component. However, in 1964 Roger Shepard published a paper "Circularity in Judgments of Relative Pitch", which described a sequence of tones in which the pitch value modulo octaves rises forever. Such a sequence is indeed perceived as rising forever, even though it is completely repetitive. The basic trick in constructing these tones is that the only harmonics are those with frequencies which are multiples of the fundamental frequency by powers of 2, i.e. 1, 2, 4, 8, 16 etc. Also, the fundamental frequency is weak relative to the second harmonic. As a result, the absolute frequency of the sound is ambiguous, even though its value modulo octaves is unambiguous.

The implication of the perception of rising tones on these **Shepard scales** is that if the perceived fundamental frequency of a pitch value is ambiguous, small changes in the pitch value modulo octaves are preferentially interpreted (by the brain) as corresponding to small changes in absolute pitch value.

If small intervals modulo octaves are unambiguous in their direction, then we would expect larger intervals to be maximally ambiguous. The largest possible interval modulo octaves is half an octave, i.e. 6 semitones, also known as a **tritone**.

The **tritone paradox** refers to a phenomenon discovered by music psychologist Diana Deutsch, which is that the ambiguity in perception of direction of tritone intervals between Shepard tones is a function of absolute pitch modulo octaves, with the function being different for different individuals.[2] For each listener there is a particular position in the scale where the direction of a tritone interval is maximally unambiguous, and the ambiguity of other tritone intervals is a function of how close the notes defining those intervals

[2] *A Musical Paradox* Diana Deutsch (Music Perception 1986)

are to the maximally unambiguous tritone. For example, a given listener might have a maximally unambiguous tritone interval of $F\sharp$ to C such that change in pitch going from $F\sharp$ to C was unambiguously perceived as going upwards.

There are at least two possible interpretations of the observed pattern of ambiguity. One is that the neural representation of pitch value modulo octaves is circular, and that a particular direction in the brain is defined as being "upwards", for example as shown in Figure 11.2.

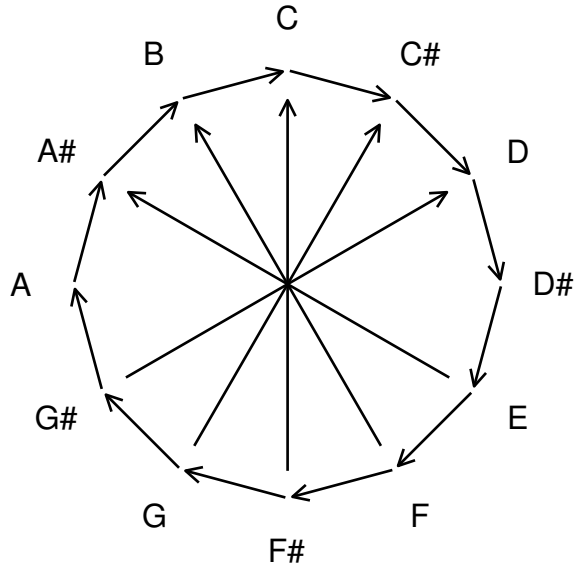

Figure 11.2. Circular tritone model. Direction for small intervals is clockwise (on the diagram) = upwards (perceived). Direction for tritones is upwards (on the diagram) = upwards (perceived). The tritone interval $F\sharp$ to C is the least ambiguous in its direction (definitely upwards); the interval A to $D\sharp$ is the most ambiguous (it could be either up or down).

A second possible interpretation is that the neural representation of pitch value modulo octaves is linear with overlap, and the maximally unambiguous tritone interval is located at the centre of this map, so that it is the least affected by ambiguous locations of neurons representing pitch values in the overlap. This interpretation is shown in Figure 11.3.

The advantage of the overlap model is that it simultaneously models perceived direction for both very small intervals and tritones. In the circular model, tritone direction is modelled by a fixed direction, whereas direction for small intervals is modelled by clockwise (or anticlockwise) motion around the circle.

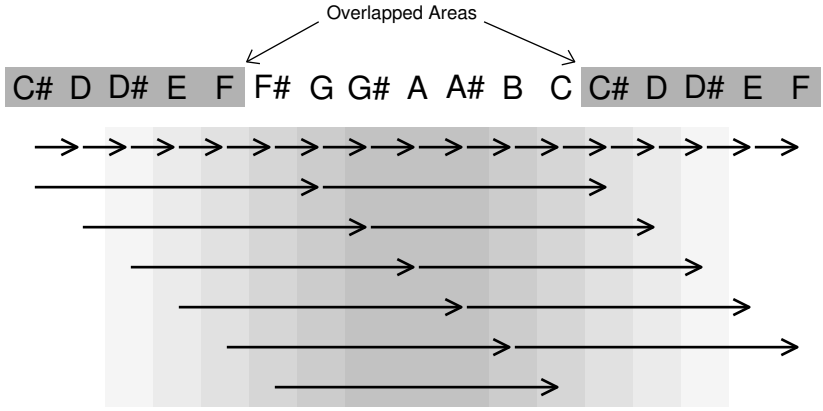

Figure 11.3. Overlap tritone model. Direction for all intervals is rightwards (on the diagram) = upwards (perceived). In this diagram only the interval $F\sharp$ to C has an unambiguous interpretation: all the others have two possible interpretations. This is a function of the size of the overlapped areas at the ends (in this case from $C\sharp$ to F). A variation on this theory is that greater priority is given to direction perceived from intervals represented in the central area of the map (as shown by the different shades of gray—darker means more weight is given to arrows lying in that region). In the example shown, $F\sharp$ to C would still be the most unambiguous upward tritone, and this would depend only on the midpoint of this interval (A) being at the centre of the map, and would not depend on how large the overlapped areas at the ends of the map were.

The other consideration making the linear overlap model more likely is that all other known cortical maps representing one-dimensional values map them in a linear fashion. In particular this applies to all known tonotopic[3] cortical maps.

The location of the maximally unambiguous tritone interval appears to be determined by the individual's exposure to spoken language, as correlations have been observed according to geographical location,[4] and also between mother and child.[5] This relationship between exposure to speech and the mechanics of octave translation invariance provides further evidence that octave translation invariance is relevant to speech perception (and not just to music perception).

[3]Reminder: a **tonotopic** map correlates position in one direction with frequency or pitch.

[4]*The Tritone Paradox: An Influence of Language on Music Perception* Diana Deutsch (Music Perception 1991)

[5]*Mothers and Their Children Hear a Musical Illusion in Strikingly Similar Ways* Diana Deutsch (Journal of the Acoustical Society of America 1996)

11.6 Octaves and Consonant Intervals

As already mentioned in Chapter 9, when discussing the relationship between invariances of pitch translation and octave translation, there is a correspondence between octave translation invariance and the importance of consonant intervals: all those aspects of music perception that depend strongly on consonant intervals are also octave translation invariant.

The one aspect of pitch perception which is not octave translation invariant, and which does not depend on perception of consonant intervals, is the perception of the up and down motion of melodic contours.

The next chapter on calibration suggests an explanation for all these observations, and also explains why octaves and other consonant intervals are so important in the first place.